Extensions of Non-negative Matrix Factorization and their Application to the Analysis of Wafer Test Data

Dissertation

zur Erlangung des Doktorgrades
der Naturwissenschaften (Dr.rer.nat.)
der Naturwissenschaftlichen Fakultät II – Physik
der Universität Regensburg

vorgelegt von

Reinhard Schachtner
aus Reischach

Februar 2010

Die vorliegende Dissertationsschrift entstand während einer dreijährigen Zusammenarbeit mit der Firma Infineon Technologies AG Regensburg.

Wissenschaftliche Betreuer:

Prof. Dr. Elmar W. Lang
Naturwissenschaftliche Fakultät III
Institut für Biophysik und physikalische Biochemie
Computational Intelligence and Machine Learning Group
Universität Regensburg

und
Dr. Gerhard Pöppel
Principal, Data Analysis
Methods Group PTE 4
Infineon Technologies AG Regensburg

Kolloquium: 23.04.2010

Vorsitzender:	Prof. Dr. Josef Zweck
1. Gutachter:	Prof. Dr. Elmar Lang
2. Gutachter:	Prof. Dr. Ingo Morgenstern
weiterer Prüfer:	Prof. Dr. Klaus Richter

Contents

Chapter 1

Introduction

The main topic of this thesis is the investigation of a recently developed machine learning technique called Non-negative matrix factorization (NMF) and its potential use for failure analysis in semiconductor industry.

1.1 Blind Source Separation and Matrix Factorization

Due to exploding data storage capacities and still improving measurement procedures, high-dimensional data are more and more widespread in diverse fields such as medicine, biology, logistics, information technology and industry. However, the rising flood of information and data complexity can rapidly become unmanageable and necessitates suitable data preprocessing algorithms in order to reduce data dimension, and to extract, visualize or encode the desired information. The purpose of low dimensional representations of a high dimensional datasets with a minimum loss of information is to uncover the most relevant features of the data. Many procedures used for dimensionality reduction can be formulated as a matrix factorization. A $N \times M$ matrix $\mathbf{X}$ is approximated by some function of the product of two smaller matrices $f(\mathbf{W}, \mathbf{H})$, where $\mathbf{W}$ has dimension $N \times K$ and $\mathbf{H}$ has dimension $K \times M$, $K << \min(M, N)$. Prominent examples of such exploratory matrix factorization (EMF) techniques represent principal component analysis (PCA) which was first proposed by Pearson [Pea01] and Hotelling [Hot33], independent component analysis (ICA) [Com94], [HKO01], [CA02] or non-negative matrix (NMF) and tensor (NTF) factorization [CZPA09]. In recent years, NMF has gained growing popularity over competing techniques such as PCA or ICA due to the direct and intuitive interpretability of the components in a parts-based representation.

1.1.1 Blind Source Separation

Rather than detecting correlations between measurement variables by statistical tests, which is a usual task e.g. in Social Sciences, Blind Source Separation techniques aim to form a quantitative model of what was observed by stating suitable assumptions on the data generative process.
Multiple cause or latent variable models assume the data to be generated by some process which is not observable directly, in which a set of, say K hidden sources superimpose and generate a set of N observations. Based on careful assumptions on the underlying process, the superposition principle is modeled and the hidden sources as well as their contributions to the observations can be reconstructed in some cases. When applied to a BSS problem, for example PCA or ICA assume that the underlying sources are mutually uncorrelated or independent, respectively.
In contrast, NMF exploits the non-negativity as a basic property of many kinds of real world data.

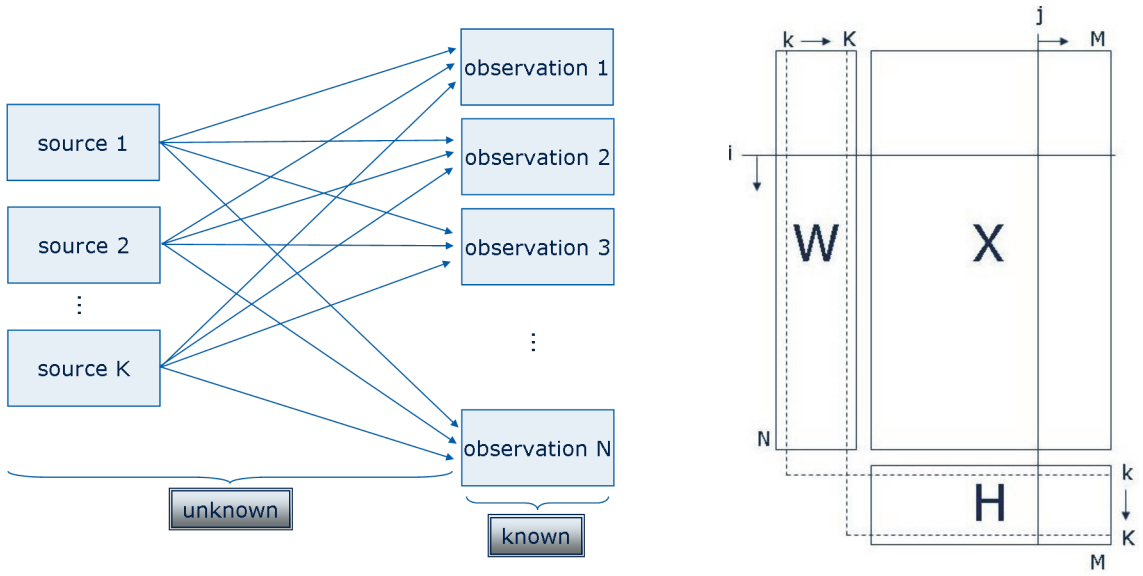

Figure 1.1: *left*: A typical task for machine learning is the Blind Source Separation problem. A set of K sources is superimposed in some way and generates a set of N observations. Each source and observation is a vector of M variables. For some reasons, the underlying sources are not observable directly, nor are the individual contributions to the single observations. *right*: Illustration of the matrix factorization model $\mathbf{X} = f(\mathbf{W}, \mathbf{H})$. A $N \times M$ data matrix $\mathbf{X}$ is approximated by a function of the product of a $N \times K$ weight matrix $\mathbf{W}$ and a $K \times M$ matrix $\mathbf{H}$. Properties related to running index i, directly transfer from $\mathbf{X}$ to $\mathbf{W}$, while index j is common for $\mathbf{X}$ and $\mathbf{H}$

1.1.2 Matrix Factorization

Under some linearity assumptions, the BSS model shown in figure 1.1 can be expressed as a matrix equation

$$\mathbf{X} \approx \mathbf{WH} \tag{1.1}$$

where the i-th observation is the M-dimensional row vector

$$\mathbf{X}_{i*} \approx \sum_{k=1}^{K} W_{ik}\mathbf{H}_{k*} \tag{1.2}$$

and can be approximated by a linear combination of the K hidden basis components $\mathbf{H}_{k*}$ using the weights W_{ik} which reflect the respective presence of component k in observation i (see Fig. 1.1 for an illustration).

1.2 Industrial data

In a microchip factory, various different products like sensors or mobile components or other integrated circuits based on semiconductor technology are manufactured in sophisticated sequences of up to several hundreds of individual processing steps.

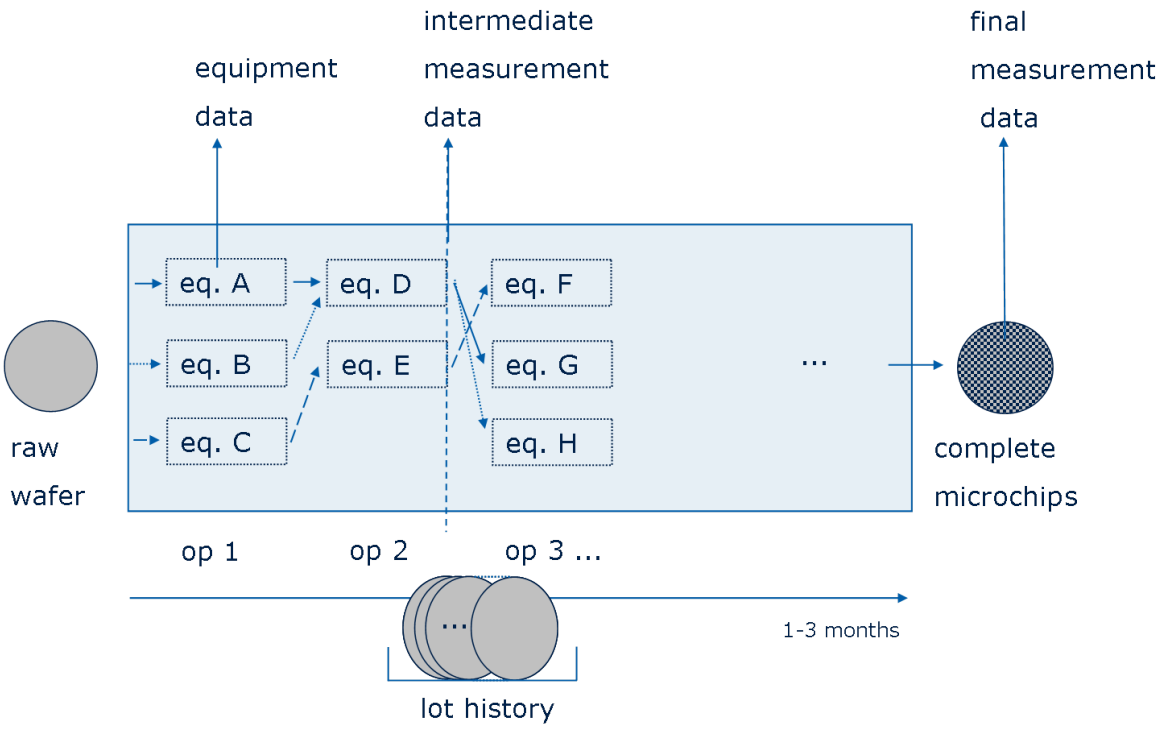

Figure 1.2: Wafer manufacturing and the various kinds of data collected during the process. Before the microchips are completed, a wafer passes up to several hundreds of single operations. Typically, wafers are bundled to units of 25 or 50 pieces called lot which share the same processing history. In general, an operation is not executed by only one single apparatus, but can be performed on several pieces of equipment in parallel. During the whole processing chain which takes 1-3 months, various kinds of data are collected: The lot history collects the information about which single equipments have processed a certain lot in the successive operations. Equipment data contains general information on the operational characteristics of the manufacturing tools. After important processing steps, intermediate measurements are performed to control the success of the preceding steps and finally, after completion, final measurements such as functional tests take place.

1.2.1 Wafer fabrication

The usual basic material consists of a silicon disk called *wafer*, which undergoes a series of processing steps called *operations*. Depending on the product, each wafer contains up to a few thousand *chips* whose manufacturing process can take 1-3 months before completion.

Each type of product has an individually specified sequence of *operations* that have to be carried out on the disk. An operation can be e.g. adding a layer to the wafer, drawing a pattern, covering a pattern with a photo layer, etching a pattern or implanting [vZ90].

Each operation follows a specific *recipe* containing details such as the thickness of a layer to be added.

In general, more than one single piece of equipment can perform an operation, e.g. there are several implanters or etchers at a semiconductor site which can perform the same operations (see figure 1.2).

Wafers are processed as a unit called *lot*, which is a collection of 25 or 50 wafers. All wafers within a lot typically share the same processing *route*, which is a certain sequence of operations and recipes.

In a wafer factory, various kinds of data arise such as

- Lot history
  For each lot, the complete path through the production line is logged, containing information about which equipment has been passed in each operation.

- Equipment data
  During production single manufacturing equipments permanently record up to 1000 operational process parameters such as e.g. temperature or pressure

- Intermediate measurement data
  After important processing steps, intermediate measurements control properties of the previous step such as a layer thickness or a line width. These data are also called *metrology* [SHLL05].

- Final measurement data
  After processing, certain test structures between the chips called process control monitors (PCM) are evaluated for each wafer , including e.g. operational voltages or currents (about 150 parameters) and functionality tests are performed on the single chips which ensure that all functional specifications are met and the chip works properly.

Functional tests which evaluate the performance of the individual chips can only be executed after the last step of production (e.g. since there are no bonds before).
Depending on the actual product, there is an individual tailored set of specifications for desired and necessary properties. According to these specifications, a series of up to 400 tests is performed, ensuring that those chips are sorted out which do not meet all required quality aspects and cannot be delivered to customers.
Each chip is labeled according to the test outcome. The labels are called *BINs* and collect certain tests. If a chip is labeled *pass*, everything works properly. If it fails in any of the tests performed, the chip carries the label of the test's BIN category (e.g. $1, 2, 3, \dots$). The BIN number codes a certain aspect of malfunction.
For each tested wafer, a *wafermap* can be generated, which is an image containing all chips with their x-y-coordinates and their BIN label.
The *yield* is defined as the portion of pass chips and is one of the most important variables in industry. According to [BC99], more than 50% of all yield problems are caused by process equipment.
Many interrelated factors affect the yield of fabricated wafers. In an unsolved low-yield problem, a series of experiments is performed by the engineers in which a set of different factors is varied to study their influence on the fabrication process. Since the number of required experiments grows exponentially with the number of varied factors, it is necessary to change as few factors as possible. This is known as *design of experiment* approach [D.91].
During wafer fabrication, large amounts of data monitoring individual processes, equipments, etc. (see figure 1.2) are stored in databases. So much effort is spent on the discovery of potential failure causes based on these data. Due to the high complexity and immense amounts of data, the need for suitable sophisticated data analysis tools in semiconductor industry is obvious.

1.2.2 Knowledge discovery by machine learning techniques

The catchword *knowledge discovery* in general means the extraction or generation of potentially useful information from data [SHLL05]. Usually, large amounts of data need to be reduced into a manageable form which can then be further processed by human experts. There is a variety of such techniques implemented in commercial software, also known as *data mining* or *automatic pattern extraction*. Data mining greatly reduces the time needed to analyze data and reveals insight extracted from data which would be otherwise unknown.

Various strategies for knowledge discovery have been applied in order to infer possible causes of faults from the data, many of which are in every-day use by semiconductor engineers.
[Tur95] investigates the necessary transformation of raw semiconductor manufacturing data into a useful form as input to a decision tree algorithm and denotes it *data engineering*.
The effectiveness of mining association rules and decision trees in determining the causes of failures of a wafer fabrication process is discussed in [BCC99]. The authors conclude that choosing a proper data mining method and a suitable data pre-processing to solve a problem at hand is a non-trivial problem and articulate the need for general methodologies and guidelines.
Due to the high data complexity, the application of neural networks has been studied in several facets. The main applications include the modeling of individual processes and process sequences, real time equipment controlling as well as failure detection [Fed96], [FPW97], classification and diagnosis of equipment and process problems [KWL$^+$97] , [GB00a] [Sch02] and the references therein.
Yield improvement strategies by statistical data analysis are discussed e.g. in [TKGB99], [BC99], [CWC07] The methods include statistical techniques like analysis of variance (ANOVA), nonparametric tests, multiple regression, generalized linear models (GLM) and K-means clustering .
A classification and regression tree (CART) method for analyzing wafer probe data was discussed in [SMR$^+$02] and compared to classical multivariate statistical techniques such as clustering, principal components and regression-based methods.
Other recently developed machine learning concepts find application in the semiconductor industry. Examples are independent component analysis (ICA), which has been used to model and analyze wafer probe test data [DPGK05], and process disturbances [SM07], or support vector clustering on wafer binary maps [Wan09]
Further, Projection Pursuit [RPW06], and mixture models combined with Bayesian networks [SPST07] have been proven to be very useful tools to discover dependencies and irregularities in multivariate data sets.
Multivariate techniques such as principal components analysis (PCA), multiway PCA and other multilinear factorizations have been applied successfully for process monitoring (see [WGB$^+$99] for the monitoring of an etching process).
Recently, ICA and PCA have also found application to semiconductor industry for multivariate process monitoring (see [LQL06], [GS07], [CHCL09] and references therein).
Non-negative matrix factorization (NMF) which will be the main topic of this thesis has recently been applied in the semiconductor industry for endpoint detection in plasma etching [RMRM08].
Summarizing, a huge arsenal of different data analysis and machine learning techniques can be applied to various kinds of data collected during and after microchip manufacturing. Any of the mentioned methods can make a contribution to the detection of failure causes and improvement of production. However, all techniques also have a limited field of application and eventually need a proper data preprocessing and skilled personnel to place the results into context and to draw viable conclusions.

1.3 This thesis

During the last three years I worked with the Infineon Technologies AG in Regensburg. The basic intention of my work was to figure out new data analysis tools which support the engineers in their daily work discovering potential flaws in the manufacturing process of semiconductor wafers.
The main topic of this thesis is the investigation of Non-negative matrix factorization and extensions thereof which were designed for the application to semiconductor wafer test data.
Although the main contributions of this thesis are rather theoretical aspects of Non-negative matrix factorization techniques which can be easily transferred to other application fields or areas of research, the potential of the new developments is demonstrated on test data sets as they arise in semiconductor industry.

We will utilize the potential of non-negative matrix factorizations by modeling the test data gathered at the end of wafer processing as a set of observations which is generated by several simultaneously acting, not directly observable sources. The final test data results from a joint action of failure causes. With a slight abuse of denotation, the term *superposition* (which exclusively describes the overlapping of waves in its original meaning) is used here to depict the principle of simultaneous action of several causes which are mixed up to produce a set of observations. By non-negative superposition we mean the property that one individual source can not destruct the effect of another source and the individual contributions can only overlap in a constructive fashion.
Two different kinds of data sets will be analyzed, both of which can be arranged in an $N \times M$ data matrix $\mathbf{X}$.

1. BIN data
   In the measurement procedure, each chip is labeled according to a BIN category (see Fig. 1.3, top). A wafer can be characterized by the counts of each individual BIN (see Fig. 1.3, bottom, right). In that case, each column $\mathbf{X}_{*j}$ corresponds to a BIN category while each wafer is stored in its own row $\mathbf{X}_{i*}$.

2. Pass/fail data
   Each chip is labeled *pass*, if all specifications are met properly, and *fail*, if any of the tests has a negative outcome. This data contains x-y- coordinates and pass-fail labels coded as 0/1 (see Fig. 1.3, bottom, left). Each row vector $\mathbf{X}_{i*}$ corresponds to a wafer, while each column vector $\mathbf{X}_{*j}$ is associated to the x-y- position of a chip in this case.

BIN data contains either count data or relative counts normalized by the total number of chips per wafer and is non-negative. In this case, we will see how NMF can be directly applied to decompose the data into potentially meaningful components.
On the other hand, pass/fail data is binary containing entries 0 or 1 only. In that case, NMF can not be applied directly and we will develop an extension of NMF suited for binary data called *binNMF*. Note that the term *BIN* in upper case letters denotes a specific test category, and is not to be confused with the lower case expression *bin* as in *binNMF* which means *binary* here.
We will associate a non-negative superposition principle for the generation of both, BIN and pass/fail data and investigate the potential of NMF to solve the blind source separation problem.
This thesis is organized as follows: Chapter 2 introduces the analysis technique of Non-negative matrix factorization (NMF). After a motivating example, a brief overview of existing applications of the method is given and the main implementational aspects concerning cost functions and optimization strategies are discussed. Further, some extensions and variants of NMF are outlined.
Chapter 3 focuses on one of the (to some extent) unsolved questions of NMF, given by the non-uniqueness of its solutions. Based on geometrical considerations, a *determinant criterion* is derived which allows for the characterization of a naturally "best" solution among several possible ones by a minimal volume spanned by the basis vectors. An algorithm incorporating this determinant criterion called *detNMF* is developed and simulations demonstrate its performance on artificial datasets. Further, the determinant criterion gives rise to a new and simple explanation for the phenomenon reported in the literature that a cascade of several NMF steps usually improves the quality of the solutions. Finally, the determinant criterion and the *detNMF* algorithm are briefly put into context of related work.
Equipped with the knowledge on optimal NMF solutions in a general setting, Chapter 4 presents the direct application of NMF on a non-negative dataset derived from BIN categories. The necessary data aggregation is described and the linearity assumptions on the data generating process required by the NMF model are stated. As a first approximation, the assumptions seem adequate. A volume constrained NMF algorithm is applied to generate decompositions of this data into putative non-negative sources and weights. It is illustrated that the extracted components are plausible from a

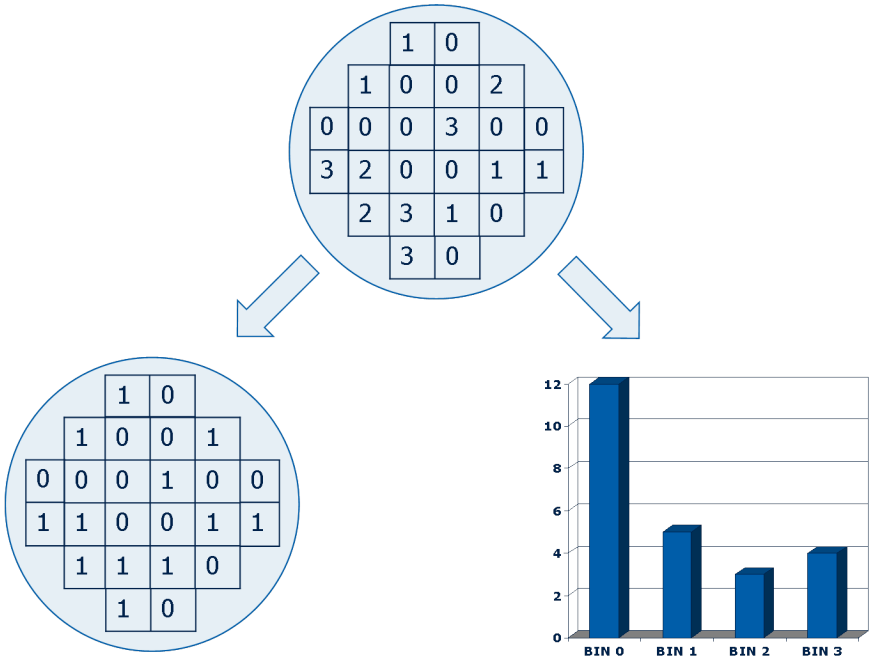

Figure 1.3: Two possibilities to generate a data matrix from a set of several wafermaps. While each row contains the data from a different wafer, the columns can represent either single chips (left) or different BIN categories. A non-negative superposition model can be associated to both cases. The example wafer contains 24 chips, each labeled by one of the BINs $0, 1, 2, 3$. Here, BIN 0 means: *die works fine*, while BIN l (l=1,2,3) codes the event *chip failed in test category* l.

data analysis point of view. Furthermore, it turns out that the determination of the actual number of underlying components can not be achieved by standard rules of thumb in general. The further analysis steps to discover the true underlying root causes are not discussed in this thesis. Such a description would be technically too complex and requires expert knowledge.
Chapter 5 presents the development of a completely new analysis technique named *binNMF*. It can be seen as an extension of NMF to binary pass/fail test data sets. The underlying superposition model is explained in detail and transformed into a maximum likelihood estimation problem. A two stage optimization strategy is developed which explores the search space in a simplified version of the problem and adds on a fine tuning procedure which optimizes the actual likelihood function. Related techniques in the literature are surveyed, concluding that none of the existing techniques covers all required aspects for this kind of investigations. A toydata example illustrates the basic properties of the *binNMF* technique, and two different real world data sets are analyzed in which interpretable components were extracted by the method. Again, the true number of underlying components can not be determined automatically.
The remaining parts of this thesis investigate the ability of Bayesian techniques to gain further insights into the NMF methodology. It is hoped that especially the Bayesian approach to model order selection and the systematic framework to incorporate prior knowledge into an estimation procedure supplements the basic NMF methods.
In a brief introduction to general Bayesian techniques in chapter 6 the basic ideas are presented. Chapter 7 reviews recent literature on attempts to incorporate Bayesian methods into NMF.
Finally, chapter 8 presents two achievements which follow from Bayesian reasoning. The first one is a theoretic description of an optimal solution to a given NMF problem in a Bayesian sense. It turns out, that under some conditions, this result derived from Bayesian arguments coincides with the determinant criterion from chapter 3. The second achievement is the development of a Bayesian extension to one of the best-known NMF algorithms. To the best of my knowledge, the algorithm in the presented form named *VBNMF* is a new development. Simulations studies demonstrate its ability to estimate the actual number of underlying components in toydata sets under certain conditions. It turns out that the *VBNMF* algorithm is relatively tolerant against slight violations of the prior assumptions and can be useful for binary datasets as well although if it is constructed for the continuous NMF case. A binary real data example further demonstrates the performance of the algorithm and highlights potential room for improvement.
Concluding, a brief summary of the main findings in this thesis and an outlook are given in chapter 9.

Chapter 2

Introduction to NMF

Non-Negative Matrix Factorization (NMF) is a multivariate data analysis technique which is aimed to estimate physically meaningful latent components from non-negative data.
Assume there are K different hidden components $\mathbf{H}_{k*}$ constituting M-dimensional row vectors of non-negative entries $H_{k1}, \ldots, H_{kM}$. As an introductory example, assume each of the M dimensions to be a different wavelength, and a hidden component $\mathbf{H}_{k*}$ to be the emission spectrum of some material. Assume further that the true components $\mathbf{H}_{k*}$ can not be measured directly for some reason. Instead, we can measure additive superpositions, such that each observation $\mathbf{X}_{i*}$ can be written as

$$\mathbf{X}_{i*} \approx \sum_{k=1}^{K} W_{ik}\mathbf{H}_{k*} \tag{2.1}$$

using the weights W_{ik} which reflect the presence of component k in observation i.
In that case the NMF procedure can be interpreted as blind source separation (BSS) challenge and consists of estimating both matrices $\mathbf{W}$ and $\mathbf{H}$ blindly, or, in an unsupervised fashion.
In some applications, the method is used to extract at least some informative components from the data (instead of a complete separation of *all* components) and separate them from the rest, e.g. in denoising tasks.
Unlike other matrix-valued blind source separation techniques such as factor analysis [Gor74], principal component analysis (PCA) [Jol86], [Pea01] , [Hot33], or independent component analysis (ICA) [Com94], NMF is especially suited for the decomposition of non-negative data. The absence of negative values is an important property of many physical quantities, such as for example intensities, concentrations, frequencies, counts and not least probabilities. The key point is to form a quantitative model of what was observed rather than detecting correlations between measurement variables [PT94].

2.1 Non-negative superposition: an example

Many real world data sets are generated by a strictly additive superposition of non-negative source values to non-negative observation data. For illustrative purposes, we discuss an air pollution study as introductory example.
Assume there are three different air pollutants A,B,C (maybe something like Nitrogen Oxides, Carbon Monoxide, Sulfur oxide, e.t.c.) and only two possible sources of pollution or contaminators, e.g. a factory and the traffic on a nearby highway.
We make the simplifying assumption that each of the sources emits a characteristic mixture of A,B and C (see figure 2.1, left) which are up-and down regulated as an ensemble. The overall emission of

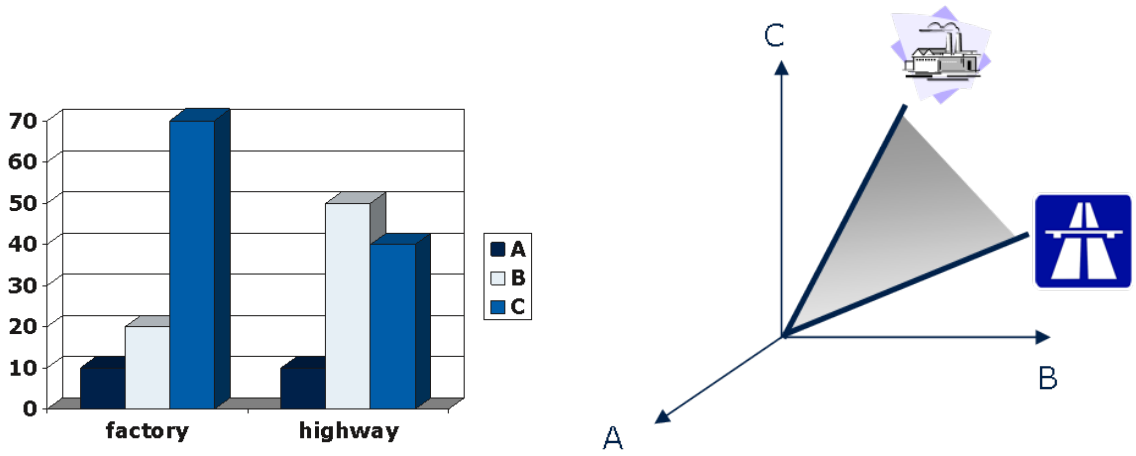

Figure 2.1: Virtual air pollution study: Assume two different sources of air pollution, each emitting a characteristic mixture of three pollutants A,B,C (left). In absence of further sources, the measurement data will lie in a 2-dim subset of 3-dim space (right)

a source can be strong or weak (e.g. rush hour or less traffic), but the proportions between the three quantities are fixed and constitute a typical fingerprint of the source. We further assume that the measured quantities originate only from the two mentioned sources and there is no other contaminator. Several air samples are taken by some apparatus which can measure quantities given by the three variables A,B, and C.
When plotting the measurement values in a 3-dimensional coordinate system, whose axes are given by A,B and C, the data will lie in a 2-dimensional subset (if there are no other sources) as illustrated in figure 2.1, right.
The locations of the measurement values can be understood as follows: If only one source is active (e.g. if the factory is temporarily down to clean the chimney), the samples taken during that time contain molecules originating exclusively from the traffic on the highway. The corresponding data lies on the lower boundary line in the figure. Since we assume that the emission profile stays constant independent from the intensity, samples corresponding to more traffic lie apart from the origin but still on the same line. If the factory starts emission, the air particles from the factory will superimpose with these from the highway. Assuming that the components do not volatilize with different speeds, the measurement values will lie somewhere in between the lines corresponding to the highway and the factory. In case of no traffic, the measured particles will come from the factory only and hence the data will lie on the upper boundary in the figure.
Expressing this as a matrix factorization model, the source profile of the factory can be described as a vector $\mathbf{H}_{1*} = (H_{11}, H_{12}, H_{13}) \geq 0$ representing the concentrations of the substances A,B and C. Similarly, the highway is assigned a basis vector $\mathbf{H}_{2*}$. Each measured data point constitutes a vector $\mathbf{X}_{i*} = (X_{i1}, X_{i2}, X_{i3})$ and can be written as

$$\mathbf{X}_{i*} = \sum_{k=1}^{K} W_{ik}\mathbf{H}_{k*} = W_{i1}\mathbf{H}_{1*} + W_{i2}\mathbf{H}_{2*} \tag{2.2}$$

where W_{i1} and W_{i2} describe the activation of the two sources in observation i.
If the actual profiles (figure 2.1, left) of neither highway nor factory were unknown, a Blind Source Separation task would be to estimate both the matrix of basis profiles $\mathbf{H}$ and individual weights $\mathbf{W}$

from the observations $\mathbf{X}$ only. In that case Non-negative matrix factorization would help to recover both matrices. The assumptions are given by the linear model in eq. 2.2 and the non-negativity constraints $\mathbf{W} \geq 0$, $\mathbf{H} \geq 0$.

2.2 Applications for NMF

The decomposition of large scale datasets of non-negative quantities plays a key role several application fields. Since its invention by Paatero and Tapper [PT94] in the mid 1990s under the name *positive matrix factorization* (PMF), dozens of researchers applied NMF to a variety of different problems in various research areas. Although not being the very first to apply NMF, Lee and Seung established the now popular name **non-negative matrix factorization** (NMF) and their *nature* publication in 1999 [LS99] initiated the nowadays great popularity of this technique in interdisciplinary working fields.
By now, variants of NMF have found application to

- Environmental Science and chemometrics, where spectra or sets of concentrations were analyzed [PT94], [Hop00] and meteorological applications [ST09]
- Image data processing, [LS01], [CGLZ01], [BB02], [LHZC03], [GVS03], [GV03] [Hoy04] where relevant parts of images are to be extracted and especially emphasis lies on face detection [BP04a] and image classification problems such as handwritten digit recognition.
- Biomedical applications such as e.g. microarray data analysis, [BTGM04], [GC05], [CSPMT$^+$06], [KP07], [SLK$^+$08], where the non-negative data corresponds to gene expression levels and NMF was successfully applied for cancer classification; Other biomedical applications include EEG signal separation and classification [LZL04], [CCR06], the investigation of protein fold recognition [OP06] and imaging disciplines such as magnetic resonance imaging (NMR) [SDB$^+$04], [DMSS07], positron emission tomography (PET) [LLCL01], or fluorescence spectroscopy [GPH04]
- Text mining [Ber01], language modeling [NM01], and document clustering [XLG03], [SBPP06] where NMF was used to extract topics or semantic features from collections of documents.
- Sound recognition and classification [SB03], [SS05], [Asa05], [CC05] where acoustic features are extracted from sound recordings yielding e.g. instrument-specific patterns and solving the acoustic source separation problem.
- Financial data [DRdFC07], to identify underlying trends in stock market data

amongst other applications such as the clustering of Scotch Whiskies [YFH09], cryptography [XYF08] or the analysis of an etching process [RMRM08]. (see also [CZPA09] and [Dev08] for recent surveys on general and biomedical applications of NMF).

2.3 The NMF problem

NMF is an unsupervised learning algorithm which decomposes a non-negative matrix $\mathbf{X}$ into a product of two non-negative matrices $\mathbf{WH}$. Here the original matrix has N rows storing the observations and M columns representing the dimensions. The two factor matrices $\mathbf{W}$ and $\mathbf{H}$ have size $N \times K$ and $K \times M$ respectively.
The NMF problem can be stated as follows: Given a non-negative $N \times M$ data matrix $\mathbf{X}$, find two non-negative matrices $\mathbf{W}$ and $\mathbf{H}$, such that their product approximates the original matrix

$$\mathbf{X} \approx \mathbf{WH} \tag{2.3}$$

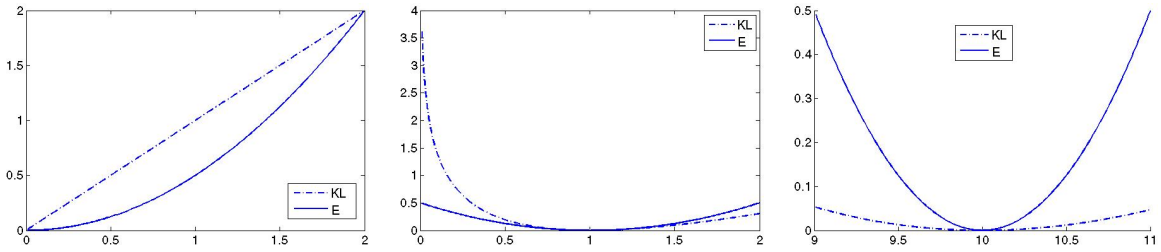

Figure 2.2: Examples for the different penalizations in the respective terms of eqns. (2.4) and D_{KL}(2.5). The x-axis denotes the term $\mathbf{WH}_{ij}$, while the y-axis shows $\frac{1}{2}(X_{ij} - [\mathbf{WH}]_{ij})^2$ (E) and $(X_{ij} \ln\left(\frac{X_{ij}}{[WH]_{ij}}\right) + [WH]_{ij} - X_{ij})$ (KL) as a function of $\mathbf{WH}_{ij}$ for three different fixed targets $X_{ij} = 0$ (left), $X_{ij} = 1$ (center) and $X_{ij} = 10$ (right).

2.3.1 Cost functions for NMF

A cost function quantifies the approximation error between $\mathbf{X}$ and $\mathbf{WH}$ and has to be minimized w.r.t. the non-negativity constraints. Popular cost functions are the squared Euclidean distance or quadratic error

$$D_E(\mathbf{X}, \mathbf{WH}) = \frac{1}{2} \sum_i \sum_j (X_{ij} - [\mathbf{WH}]_{ij})^2 \tag{2.4}$$

or the generalized Kullback-Leibler divergence

$$D_{KL}(\mathbf{X}, \mathbf{WH}) = \sum_i \sum_j (X_{ij} \ln\left(\frac{X_{ij}}{[\mathbf{WH}]_{ij}}\right) + [\mathbf{WH}]_{ij} - X_{ij}) \tag{2.5}$$

which were used in [LS99][LS01], but are by far not the only possible choices. These cost functions are sums of individual terms for each entry of the data matrix X_{ij}. Different cost functions assign different penalizations for missing the target values (see figure 2.2 for a comparison of the single entries constituting D_E (2.4) and D_{KL}(2.5) for different target values.)
There is a large variety of divergence measures which have been successfully applied to NMF problems, such as the family of Bregman divergences [DS05], Renyi's information measure [Dev05], Csiszar's divergences [CZA06], Kompass' divergence [Kom07], the α-divergence [CLKC08] or the Itakura-Saito divergence [FBD09] (see also [CZPA09] for an extensive review) which will not be discussed here.

2.3.2 Optimization strategy for NMF

The minimum of a cost function D given by e.g. eq. (2.4) or eq. (2.5) w.r.t. $\mathbf{W} \geq 0$ and $\mathbf{H} \geq 0$, can not be computed in closed form but must be found by an iterative procedure alternating between updating one matrix while keeping the other one fixed (see figure 2.3).
For example, Lee and Seung [LS99][LS01] proposed the following multiplicative update rules for the quadratic cost function D_E (2.4)

$$H_{kj} \leftarrow H_{kj} \frac{[\mathbf{W}^T\mathbf{X}]_{kj}}{[\mathbf{W}^T\mathbf{WH}]_{kj}} \quad \text{and} \quad W_{ik} \leftarrow W_{ik} \frac{[\mathbf{XH}^T]_{ik}}{[\mathbf{WHH}^T]_{ik}} \tag{2.6}$$

- `Initialize` $N \times K$ `matrix` $\mathbf{W} \geq 0$ `and` $K \times M$ `matrix` $\mathbf{H} \geq 0$
- `Repeat`
  {
  1. `Update` $\mathbf{W}$ `s.th.` $D(\mathbf{X}, \mathbf{W}^{new}\mathbf{H}) \leq D(\mathbf{X}, \mathbf{W}^{old}\mathbf{H})$ `and` $\mathbf{W}^{new} \geq 0$ `keeping` $\mathbf{H}$ `fixed`
  2. `Update` $\mathbf{H}$ `s.th.` $D(\mathbf{X}, \mathbf{W}\mathbf{H}^{new}) \leq D(\mathbf{X}, \mathbf{W}\mathbf{H}^{old})$ `and` $\mathbf{H}^{new} \geq 0$ `keeping` $\mathbf{W}$ `fixed`

  }
  `until convergence`

Figure 2.3: Pseudo code for an NMF algorithm

and the rules (2.7) for the generalized KL-divergence D_{KL}(2.5)

$$H_{kj} \leftarrow H_{kj} \frac{\sum_i W_{ik} X_{ij} / [\mathbf{WH}]_{ij}}{\sum_l W_{lk}} \quad \text{and} \quad W_{ik} \leftarrow W_{ik} \frac{\sum_j H_{kj} X_{ij} / [\mathbf{WH}]_{ij}}{\sum_p H_{kp}} \tag{2.7}$$

and proved that these multiplicative updates never increase the respective cost function. We will discuss several alternative optimization strategies in section 2.4.
The initialization of the matrices $\mathbf{W}$ and $\mathbf{H}$ can be arbitrary non-negative numbers. Several pre-processing strategies have been tested, such as PCA or clustering (see e.g. [LMA06], [ZYZ07]). In general, NMF algorithms are gradient based techniques and hence suffer from getting stuck in local minima and it is recommended to run an algorithm several times using different (e.g. random) initializations.
The convergence properties depend on the particular algorithm. In general, global convergence is hard to prove, but most algorithms can be shown not to increase the cost function in each single iteration. Usually, convergence is tested by measuring the decrease of the cost function between successive iterations, and the algorithm stops if the decrease falls below some predefined threshold [GS00], [Lin07a], [KSD08], [KP07], [KP08].
Basically, the following questions need to be considered on an NMF application:

1. Cost function
   First of all, a cost function has to be defined which expresses the discrepancy between the matrices $\mathbf{X}$ and the product $\mathbf{WH}$. There are several possible choices of a cost function for NMF (see paragraph 2.3.1).

2. Additional constraints
   In many applications there is some knowledge about the desired properties of the factor matrices available such as spareness or smoothness. Such constraints are closely related to the uniqueness of solutions, which are discussed in section 2.5.1.

3. Optimization algorithm
   Once the cost function including optional additional constraints is chosen, there are various possible numerical strategies to solve the optimization problem. A brief overview of optimization strategies in NMF is given in section 2.4.

4. Dimensionality
   The $N \times M$ matrix $\mathbf{X}$ can be factorized into the product of a $N \times K$ and a $K \times M$ matrix.

The factorization rank K is usually chosen $< min(N, M)$ and required as input for most NMF algorithms. In many real world applications, the actual number of components is unknown and its determination is a difficult issue called *model order selection*. We will discuss that question in a later paragraph (section 8.2). By then we assume the number of components K to be given.

2.4 Algorithms for NMF

In this section we will discuss several different optimization strategies which have been proposed to solve the NMF problem. Good survey papers on NMF algorithms are e.g. [Tro03], [CP05] ,[SD06], [BBL$^+$06] and the recently released textbook of Cichocki et al. [CZPA09].

2.4.1 Gradient approaches

We mainly concentrate on the plain Euclidean distance D_E (eq. 2.4) and demonstrate the update details for the matrix $\mathbf{H}$ only for demonstration purposes. The corresponding expressions for $\mathbf{W}$ follow from symmetry arguments by considering the transposed equation $\mathbf{X}^T \approx \mathbf{H}^T\mathbf{W}^T$. Most strategies can be extended to other cost functions and suitable additional constraints.
Consider the squared Euclidean distance cost function

$$D_E(\mathbf{X}, \mathbf{W}, \mathbf{H}) = \frac{1}{2}\sum_i \sum_j (\mathbf{X}_{ij} - [\mathbf{WH}]_{ij})^2 \tag{2.8}$$

whose partial derivative w.r.t. the (k, j)-th entry of $\mathbf{H}$ is given by

$$\frac{\partial D_E}{\partial H_{kj}} = -\sum_i (X_{ij} - \sum_l W_{il}H_{lj})W_{ik} \tag{2.9}$$

and involves only entries from the j^{th} column $\mathbf{H}_{*j}$. In the following, we denote

$$[\nabla_H D_E]_{*j} := \left[\frac{\partial D_E}{\partial H_{1j}}, \ldots, \frac{\partial D_E}{\partial H_{Kj}}\right]^T \tag{2.10}$$

the $K \times 1$ gradient vector whose k'th entry is given by eq. (2.9).
A gradient algorithm updates the j'th column $\mathbf{H}_{*j}$ by taking a step into the into the direction of the negative gradient (which is guaranteed to be a descent direction of D_E)

$$\mathbf{H}_{*j} \leftarrow \mathbf{H}_{*j} - \alpha_j[\nabla_H D_E]_{*j} \tag{2.11}$$

where $\alpha_j > 0$ is a step size parameter which has to be chosen suitably such that none of the following cases appears

1. α_j too small
   The decrease of the objective function is very small and the algorithm will be slow

2. α_j too large
   Either the algorithm passes the minimum and leads to a larger cost than the initial one $D_E(H_{*j}^{new}) > D_E(H_{*j}^{old})$, or H_{*j}^{new} contains undesired negative values, or both

Several NMF algorithms have been proposed which can be seen as (sometimes rescaled) gradient algorithms. In general, one can distinguish between those algorithms which solve the constrained optimization problem in each sub step exactly and set $\mathbf{H}^{new}$ to that matrix which minimizes D_E for a given $\mathbf{W}$ and those which just decrease the cost function in each sub step.

The Lee and Seung algorithm and variants

Lee and Seung [LS99],[LS01] proposed the following multiplicative update rule

$$H_{kj} \leftarrow H_{kj} \frac{[\mathbf{W}^T\mathbf{X}]_{kj}}{[\mathbf{W}^T\mathbf{W}\mathbf{H}]_{kj}} \tag{2.12}$$

which can be interpreted as a gradient step after rewriting it in an additive form:

$$H_{kj} \leftarrow H_{kj} + \eta_{kj} \left([\mathbf{W}^T\mathbf{X}]_{kj} - [\mathbf{W}^T\mathbf{W}\mathbf{H}]_{kj} \right) \tag{2.13}$$

where the step size is given by

$$\eta_{kj} = \frac{H_{kj}}{[\mathbf{W}^T\mathbf{W}\mathbf{H}]_{kj}} \tag{2.14}$$

and we recognize the last expression in (2.13) to be the negative gradient

$$[\nabla_H D_E]_{kj} = -\sum_i (X_{ij} - \sum_l W_{il}H_{lj})W_{ik} \tag{2.15}$$

Note that, in contrast to the basic gradient step (eq. 2.11), each entry H_{kj} has an extra step size parameter η_{ij} instead of one parameter per column α_j.
Lee and Seung provided a proof that an update of the form (2.12) never increases the cost function (2.8). Since all factors on the right hand side of eq. 2.12 are non-negative, the non-negativity constraint is incorporated automatically in an elegant way.
(Note that this algorithm intrinsically was known before as image space reconstruction algorithm (ISRA) [DWM86], [DP87] which was designed for a known matrix $\mathbf{W}$ and unknown $\mathbf{H}$. Lee and Seung [LS99],[LS01] were the first to use this algorithm switching between updates for $\mathbf{W}$ and $\mathbf{H}$. They also proposed a multiplicative algorithm for the generalized KL divergence 2.5 (see eq. 2.7), which was known as the Richardson-Lucy algorithm [Ric72],[Luc74],[LC84] in a similar fashion [CZPA09]).
Some problems with the Lee and Seung algorithms have been encountered. First, once a parameter is exactly zero, it remains zero for all times. Second, the convergence proof does not preclude saddle points (this was criticized by several authors, see e.g. [CDPR04], [GZ05], [BBL$^+$06]). Thus, improvements to the original algorithm have been proposed, such as the interior-point gradient (IPG) method [MZ05], which uses the modified update

$$H_{kj} \leftarrow H_{kj} + \alpha_j \eta_{kj} \left([\mathbf{W}^T\mathbf{X}]_{kj} - [\mathbf{W}^T\mathbf{W}\mathbf{H}]_{kj} \right) \tag{2.16}$$

where α_j can be chosen in closed form as large as possible such that neither the non-negativity constraint nor the decrease property is violated. This modification encourages larger step sizes and was shown to be significantly faster than the original version.
An other refinement of the Lee/Seung algorithm modified the step size (2.14) to

$$\eta_{kj} = \frac{\overline{H}_{kj}}{[\mathbf{W}^T\mathbf{W}\overline{\mathbf{H}}]_{kj} + \delta} \quad , \quad \overline{H}_{kj} =: \begin{cases} H_{kj} & \text{if } [\nabla_H D_E]_{kj} \geq 0 \\ \max(H_{kj}, \sigma) & \text{if } [\nabla_H D_E]_{kj} < 0 \end{cases} \tag{2.17}$$

where δ and σ are pre-defined small positive numbers and proved that any limit point of the modified updates is a stationary point [Lin07a], which is a necessary condition for a local minimum.

Projected Gradient methods

In contrast to interior-point methods which never leave the non-negative orthant, projected gradient methods follow a different strategy. They perform a step into some descent direction and then project the result back onto the non-negative orthant. This procedure implies the problem that the projected

step actually increases the cost function, although the same step without projection would decrease the (unconstrained) cost function (see [KSD07] and the illustration in figure 2.4). We note that this problem presumes that the actual solution lies in the non-negative orthant. In NMF problems which do have an exact solution $\mathbf{X} = \mathbf{WH}$, the final solution has no negative entries. During the optimization procedure, however, the best suiting unconstrained $\mathbf{H}$ given some intermediate $\mathbf{W}$ can indeed contain negative entries and the problem can appear.

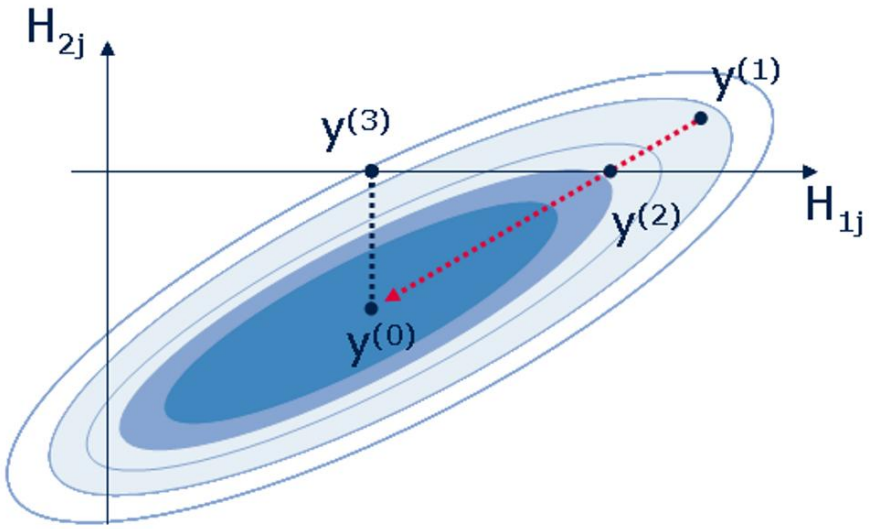

Figure 2.4: 2-dimensional example in which the variables H_{1j}, H_{2j} are updated by a projected gradient method from their initial values $y^{(1)}$. The ellipses denote level sets of the cost function D. While the unconstrained optimum $y^{(0)}$ is forbidden due to the non-negativity constraints, and the constrained optimum is at $y^{(2)}$, the projection of the unconstrained optimum $y^{(3)}$ lies on a higher level of the cost function than the initial values $y^{(1)}$ (example adapted from [KSD07])

One step of a projected gradient method can be described as

$$\mathbf{H}_{*j} \leftarrow P\left[\mathbf{H}_{*j} - \alpha_j [\nabla_H D]_{*j}\right] \tag{2.18}$$

where

$$P[z_i] = \begin{cases} z_i & \text{if } l_i < z_i < u_i \\ u_i & \text{if } z_i \geq u_i \\ l_i & \text{if } z_i \leq l_i \end{cases} \tag{2.19}$$

is a projection operator which truncates too small and too large values and sets them to predefined constant values l_i and u_i [Lin07b].
There are different variants of projected gradient methods which use different strategies to chose a proper step size α_i, such as the *Armijo rule* [Lin07b], [CZPA09].
Another type of NMF algorithms makes use of the second order derivatives and perform steps of the kind

$$\mathbf{H}_{*j} \leftarrow P\left[\mathbf{H}_{*j} - \alpha \mathbf{D} \nabla_{H_j} D\right] \tag{2.20}$$

where $\alpha > 0$ and $\mathbf{D}$ is an approximation of the inverse Hessian matrix.

There are many possible choices for $\mathbf{D}$ ranging from the identity matrix to the actual inverse of the Hessian. If $\mathbf{D}$ is the inverse Hessian, $\alpha = 1$ and no projection were applied, the algorithm is actually a Newton algorithm. Approximating the inverse Hessian instead of its exact evaluation can gain great savings of computational time. These iterations are then called *quasi-Newton* [ZC06],[KSD06] or *Newton-type* [KSD07].
Interior point methods and the projected gradient strategy can also be combined by dividing, at each iteration, the variables into a *free* and a *fixed* set [KSD07] . The latter is given by those variables, for which any step into the direction along the negative gradient would lead to a violation of the non-negativity constraint. These variables are not updated in the corresponding iteration. In every single iteration, the set of fixed variables must be identified and only the free variables are updated via eq. (2.20).
A variety of other projected Gradient approaches have been applied to the NMF problem such as *Oblique Projected Landweber*, *Barzilai-Borwein Gradient Projection*, *Projected Sequential Subspace Optimization* among others and I refer to [ZC08], [CZPA09] for more details. In summary, projected gradient approaches to NMF are more efficient than their multiplicative counterparts [CLKC08].
However, the implementation details which are necessary to assure a decreasing cost function under the non-negativity constraints seem to be rather complex. Especially if additional constraints have to be incorporated or even alternative cost functions are necessary which better suit the problem, potentially large effort has to be taken to customize a suitable algorithm.

2.4.2 Alternating Least Squares algorithms

While the principles discussed in section 2.4.1 can be applied to arbitrary NMF cost functions D, we will introduce a technique known as Alternating Least Squares (ALS) here. As suggested by the name it is exclusively valuable for the least squares ($\hat{=}$ minimized Euclidean distance) cost function D_E.
We can rewrite the squared Euclidean distance functional (eq. 2.8) column per column

$$D_E = \sum_{j=1}^{M} \left[\frac{1}{2} \sum_{i=1}^{N} \left(\mathbf{X}_{ij} - [\mathbf{WH}]_{ij} \right)^2 \right] \tag{2.21}$$

and gain a sum of M vector-valued subproblems of the form

$$\text{Minimize } f(\mathbf{x}) = \frac{1}{2} \sum_{i} \left([\mathbf{Ax}]_i - \mathbf{b}_i \right)^2 \tag{2.22}$$

subject to

$$\mathbf{x} \geq 0 \tag{2.23}$$

where $\mathbf{A} = \mathbf{W}$, $\mathbf{b} = \mathbf{X}_{*\mathbf{j}}$ and $\mathbf{x} = \mathbf{H}_{*j}$.
This vector-valued constrained optimization problem can be solved e.g. by the Lawson and Hanson procedure [LH74],[LH87].
A simple NMF algorithm can be implemented in Matlab e.g. using the function `lsqnonneg.m`, which is an implementation of the Lawson and Hanson procedure (see fig. (2.5) for the pseudo-code). Lin [Lin07b] proves the convergence of the above procedure. This property is not trivial, since, even if each sub-problem update of one column of $\mathbf{H}_{*j}$ or update of one row of $\mathbf{W}_{i*}$ is a convex problem and thus has a unique optimum [LH87], this is not sufficient for the whole problem. By the aid of the work of Grippo and Sciandrone [GS00], the convergence of this 2-block coordinate descent algorithm with one block being $\mathbf{W}$ and the other one $\mathbf{H}$ can be shown. Though theoretically faultless, this algorithm is not attractive for applications because it is extremely slow (see [Lin07b] for numerical examples).

A basic Alternating least squares algorithm for NMF can be stated as

- initialize matrix $\mathbf{W} \geq 0$
- Repeat
  {
  1. for $j = 1, \ldots, M$ update H_{*j} by solving eq. (2.22), setting $\mathbf{A} = \mathbf{W}$, $\mathbf{b} = \mathbf{X}_{*j}$ and $\mathbf{x} = \mathbf{H}_{*j}$
  2. for $i = 1, \ldots, N$ update W_{i*} by solving eq. (2.22), setting $\mathbf{A} = \mathbf{H}^T$, $\mathbf{b} = [\mathbf{X}_{i*}]^T$ and $\mathbf{x} = [\mathbf{W}_{i*}]^T$

  }
  until convergence

Figure 2.5: NMF procedure which solves a set of vector-valued non-negative least squares problems in every sub step

In spite of solving the constrained optimization subproblems, a very popular approach is to solve the related unconstrained problem instead and then project the result onto the non-negative orthant [LMA06].
Setting the derivatives (2.9) to zero for all k, j yields

$$\sum_i X_{ij} W_{ik} = \sum_i \sum_l W_{il} H_{lj} W_{ik} \qquad \text{for all k,j} \tag{2.24}$$

which is equivalent to

$$\mathbf{W}^T \mathbf{X} = \mathbf{W}^T \mathbf{W} \mathbf{H} \tag{2.25}$$

and can be solved for $\mathbf{H}$

$$\mathbf{H} = [\mathbf{W}^T \mathbf{W}]^{-1} \mathbf{W}^T \mathbf{X} \tag{2.26}$$

by multiplying the left (pseudo-) inverse of $\mathbf{W}^T \mathbf{W}$.
In general, this matrix $\mathbf{H}$ contains undesired negative values and has to be projected onto the non-negative orthant by setting the negative entries to zero or small positive constants ϵ.
The whole projected ALS algorithm can be cast in Matlab notation (see also [CZPA09])

```
W = rand(N,K);
for iter = 1:max_iter
  H = max(eps,pinv(W'*W)*W'*X);
  W = max(eps,X*H'*pinv(H*H'));
end
```

where the command `pinv` computes the Moore-Penrose pseudo inverse [Moo20], [Pen55] in Matlab. Note that the matrices to be inverted are both sized $K \times K$ only.
While convergence of this type of algorithm is theoretically hard to prove, it is often observed in practice.

Cichocki et al. [CZPA09] call this simple and fast projected ALS algorithm their *working horse*, admitting its unstable convergence properties, and suggest its use to generate good initial values for more sophisticated NMF algorithms.
Any of the presented algorithms can only converge to a local minimum. Hence, multiple runs with different random initializations are recommended anyway.

2.4.3 Stopping criteria

Beyond a predefined number of iterations or a fixed running time, some other stopping criteria were suggested for NMF algorithms in the literature. Brunet et al. [BTGM04] use the difference between recent iterations as a stopping criterion. Lin [Lin07b] invokes a stopping criterion from bound-constrained optimization.
Kim and Park [KP08] test the convergence of a NMF algorithm by an averaged measure of violations of the Karush-Kuhn-Tucker (KKT) optimality conditions

The Karush-Kuhn-Tucker optimality conditions for NMF

The Karush-Kuhn-Tucker conditions are a generalization of the method of Lagrange multipliers and necessary for a solution in nonlinear programming to be optimal [CP07] .
Given the optimization problem

$$\text{Minimize } D(\mathbf{W},\mathbf{H}) \text{ s.t. } \mathbf{W} \geq 0, \mathbf{H} \geq 0 \tag{2.27}$$

According to the Karush-Kuhn-Tucker optimality conditions, $(\mathbf{W},\mathbf{H})$ is a stationary point of eq.(2.27) if and only if

$$\begin{aligned} \mathbf{W} &\geq 0 && (2.28)\\ \nabla_W D(\mathbf{W},\mathbf{H})) &\geq 0 && (2.29)\\ \mathbf{W} .* \nabla_W D(\mathbf{W},\mathbf{H}) &= 0 && (2.30) \end{aligned}$$

and

$$\begin{aligned} \mathbf{H} &\geq 0 && (2.31)\\ \nabla_H D(\mathbf{W},\mathbf{H})) &\geq 0 && (2.32)\\ \mathbf{H} .* \nabla_H D(\mathbf{W},\mathbf{H}) &= 0 && (2.33) \end{aligned}$$

where $.*$ means component-wise multiplication. In the strict sense, any NMF algorithm should verify the validity of the KKT conditions for the estimated solutions to ensure convergence.
Proofs of convergence of NMF algorithms are scarce (see the discussions in [SD06], [BBL$^+$06]) though for the popular class of multiplicative update algorithms convergence to a stationary point could be proven [Lin07a].
We conclude this paragraph on NMF algorithms here, noting that there are by far more optimization techniques for NMF, such as *Quadratic Programming* [ZC06] or combinations of the mentioned techniques [ZC07]. The group of A. Cichocki implemented and intensively tested a variety of numerical optimization schemes for NMF problems, some of which are contained in the non-commercial *NMFLAB* software package [CZ06].

2.5 Extensions and variants of NMF

2.5.1 Additional Constraints

Obviously, a non-negative factorization $\mathbf{X} = \mathbf{WH}$ does not have a unique solution. If we insert the $K \times K$ identity matrix $\mathbf{1}_K$,

$$\mathbf{X} = \mathbf{WH} = \mathbf{W1}_K\mathbf{H} = \mathbf{WS}^{-1}\mathbf{SH} \tag{2.34}$$

we see that if $\mathbf{W} \geq 0, \mathbf{H} \geq 0$ is a NMF-solution, then $\mathbf{WS}^{-1}, \mathbf{SH}$ is a NMF-solution as well as long as $\mathbf{WS}^{-1} \geq 0, \mathbf{SH} \geq 0$.
Thus, the plain cost function $D(\mathbf{X}, \mathbf{WH})$ measuring the discrepancy between the data and its approximation cannot distinguish between two alternative solutions. In general, this problem is attenuated by additional regularization terms to the reconstruction error $D(\mathbf{X}, \mathbf{WH})$ yielding

$$\tilde{D} = D(\mathbf{X}, \mathbf{WH}) + \alpha f(\mathbf{W}) + \beta g(\mathbf{H}) \tag{2.35}$$

where $f()$ and $g()$ are application-dependent functionals. This new constrained cost function $\tilde{D}$ distinguishes between solutions of equal reconstruction errors by means of the additional terms which describe a priori information such as smoothness or sparseness of the factor matrices. The usefulness of the additional constraints must be justified by arguments which are relevant for the application [PT94].

sparsity

In particular sparsity constraints have gained quite popularity. The concept of sparse coding means that only a few units out of a large population are actually used to represent typical data vectors [Fie94]. This implies that most units take small values and only a few take significantly non-zero values when representing a data item.
Hoyer [Hoy02] uses the following cost function for NMF

$$D_{nnsc} = \sum_i \sum_j (X_{ij} - [\mathbf{WH}]_{ij})^2 + \lambda \sum_{ik} W_{ik} \tag{2.36}$$

to penalize large entries in the weight matrix $\mathbf{W}$ and encourage sparse solutions. Hoyer [Hoy04] further showed that controlling the degree of sparseness of the component matrices leads to parts-based representations that match the intuitive features of the data.
Another approach was due to Li et al. [LHZC03] encouraging spatially localized representations of images in the local NMF (LNMF) cost function

$$D_{lnmf} = \sum_i \sum_j (X_{ij} \ln \frac{X_{ij}}{[\mathbf{WH}]_{ij}} - X_{ij} + [\mathbf{WH}]_{ij}) + \alpha \sum_{ia} [\mathbf{W}^T\mathbf{W}]_{ia} - \beta \sum_k [\mathbf{HH}^T]_{kk} \tag{2.37}$$

In this case, the Kullback-Leibler divergence is chosen for the reconstruction error, and a term penalizing large entries in $\mathbf{W}^T\mathbf{W}$ as well as a term penalizing small diagonal entries of $\mathbf{HH}^T$ are added. The term for $\mathbf{W}$ should increase the sparsity of the weights and encourage solutions where only a few basis components are necessary to represent each observation, while the last expression should make the basis vectors as orthogonal as possible in order to minimize redundancy between different basis vectors. A further example incorporating sparsity to NMF is called Non-smooth NMF [PMCK$^+$06].
The actual form of the regularization constraints depends not only on the desired characteristics of the weights and basis components, but also on the optimization strategy. For example, a sparseness measure which counts the close to zero entries induces discontinuities to the cost function and requires more sophisticated optimization strategies such as the genetic algorithm proposed in [STP$^+$05].

2.5.2 NMF extensions and related techniques

Supervised NMF

Supervised extensions of LNMF (eq. 2.37) called Discriminant NMF (DNMF) [BP04b] or Fisher NMF [WJ04] allow for incorporation of additional knowledge such as class labels and encourage similarities between observations from the same class and dissimilarities between the classes.

Positive matrix factorization

In the original *positive matrix factorization* problem [PT94]

$$\mathbf{X} = \mathbf{WH} + \mathbf{E} \tag{2.38}$$

$$Q = \sum_i \sum_j \frac{E_{ij}^2}{\sigma_{ij}^2} \tag{2.39}$$

the optimization problem is formulated as

$$\tag{2.40}$$

$$\{\mathbf{W}, \mathbf{H}\} = \text{argmin } Q \tag{2.41}$$

where the matrix σ contains error estimates of the observations. Thus, individual data points $\mathbf{X}_{ij}$ which may be uncertain are scaled by a large σ_{ij} and do not participate to the cost function as they would if all σ_{ij} were equal. NMF algorithms with special accentuation on robustness w.r.t. potential outliers thus could be easily developed by introducing additional weighting terms.

Affine NMF

Laurberg et al. [LH07] included an explicit offset and showed that this affine NMF model can lead to a more accurate identification of mixing and sources. In our notation, each observation is written as

$$\mathbf{X}_{i*} \approx \sum_{k=1}^{K} W_{ik}\mathbf{H}_{k*} + \mathbf{H}_{0*} \tag{2.42}$$

where $\mathbf{H}_{0*}$ is an additional vector. This special case can be reached by fixing one column of $\mathbf{W}$ to one during the iterations.
Affine NMF can be useful in cases where one basis component is present in every observation with an equal weight. [LH07] showed that this can help to improve the uniqueness of solutions.

Multilayer techniques

To further improve the performance of NMF algorithms and to reduce the risk to get stuck in local minima, a multi-layer technique has been proposed recently [STP+05], [CZA06], [CZ07]. A cascade of $L > 1$ decompositions of the type

$$\mathbf{X} = \mathbf{WH}^1\mathbf{H}^2 \dots \mathbf{H}^L \tag{2.43}$$

can significantly improve the results compared to one single decomposition.
While [CZ07] use this technique together with sparsity constraints, they argue that a cascade of sparse representations can lead to a non-sparse representation, which is closer to the truth in some applications. In chapter 3 we will give an alternative explanation for the success of multilayer techniques.

Bayesian NMF

Bayesian methods have gained much attention in the machine learning community in the last decades. Their potential to gain further insights into NMF was investigated recently [VCG08], [SL08], [SWH09], [Cem09], [FBD09] and is still an area of active research.
We will discuss the Bayesian approaches to NMF in more detail in a later paragraph (section 7.2).

Nonnegative Tensor Factorization

Non-negative matrix factorization can be viewed as a bilinear model. A $N \times M$ object is factorized into something of size $N \times K$ and a second factor of size $K \times M$. A similar model could be designed for something that consists of $N \times M \times L$ dimensions. e.g. N observations in M variables at L time points, or several space coordinates treated separately, or individual test categories, etc. Many datasets have to be aggregated before being representable as a matrix, and applying NMF. This aggregation can be related to a loss of information and it can be desirable to decompose the original, multilinear object instead of a bilinear matrix.
For example in image analysis, the neighborhood of pixels can be relevant information. When vectorizing each image in order to store it into a row of the data matrix $\mathbf{X}$, all pixels of the image are aligned, e.g. column by column. The information that two pixels are on a horizontal line is completely lost. Instead, it could be interesting to decompose the three way tensor built up from N observations in x- and y- coordinates, yielding a $N \times M \times 2$ -tensor.
One possibility is to decompose a $N \times M \times L$ tensor $\mathbf{X}$ via

$$\mathbf{X}_l \approx \mathbf{W}\mathbf{D}_l\mathbf{S}_l (l = 1, 2, ..., L) \tag{2.44}$$

where $\mathbf{X}_l$ is the l^{th} slice of a three-way tensor. Non-negative tensor factorizations are a natural extension to NMF and a current area of ongoing research [CZPA09].

Kernel NMF

As mentioned in [SS02], a nonlinear extension of a linear technique can be designed by means of the concept of kernels, provided that the linear technique can be expressed in terms of scalar products. Since the matrix products in NMF can be formulated via the scalar product $\langle \mathbf{x}, \mathbf{y} \rangle := \sum_k \mathbf{x}_k \mathbf{y}_k$

$$[\mathbf{WH}]_{ij} = \langle \mathbf{W}_{i*}, \mathbf{H}_{*j} \rangle \tag{2.45}$$

a kernel version of NMF should be possible by replacing the scalar product by a kernel matrix.
One such kernel NMF formulation was presented in [ZZC06] and is based on a mapping

$$\phi : x \in V \rightarrow \phi(x) \in F \tag{2.46}$$

where ϕ is a nonlinear map from the original input space V to a possibly infinite dimensional feature space F and $\phi(\mathbf{X}) = [\phi(\mathbf{X}_{1*}), \ldots, \phi(\mathbf{X}_{N*})]$. Kernel NMF is aimed to find

$$\phi(\mathbf{X}) = \mathbf{W}\mathbf{H}_\phi \tag{2.47}$$

where $\mathbf{H}_\phi$ is a basis in feature space and $\mathbf{W}$ are the usual weights.
The kernel trick implies that the mapping ϕ need not be known explicitly. Multiplying eq.(2.47) from right by $\phi(\mathbf{X})^T$ leads to

$$\phi(\mathbf{X})\phi(\mathbf{X})^T = \mathbf{W}\mathbf{H}_\phi\phi(\mathbf{X})^T \tag{2.48}$$

yielding

$$\mathbf{K} = \mathbf{W}\mathbf{Y} \tag{2.49}$$

where $\mathbf{K}$ is called a kernel matrix whose ijth entry is given by

$$\langle \phi(\mathbf{X}_{i*}), \phi(\mathbf{X}_{j*}) \rangle \tag{2.50}$$

Since $\phi(\mathbf{X}_{i*})$ lies in F for all $i = 1, \ldots, N$ and $\mathbf{H}_\phi$ is a basis of F, one can express $\mathbf{H}_\phi$ by the images $\phi(\mathbf{X}_{i*})$ to obtain a representation of $\mathbf{Y}$ in terms of the kernel matrix $\mathbf{K}$.
In some applications, the nonlinear kernel extension can extract more useful features from the data than its linear counterpart. Moreover, it can deal with relational data where only the relationships between objects are known (expressed as a kernel matrix) [ZZC06] .
Another kernel NMF method based on convex NMF was presented in [DLJar], using a linear kernel $\mathbf{K} = \mathbf{X}\mathbf{X}^T$.
The design of optimal nonlinear kernels for the detection of non-linear features in certain applications seems to be an interesting field of future investigations.

2.5.3 Related techniques

We will not review general matrix factorization techniques like singular value decomposition (SVD), principal component analysis (PCA), independent component analysis (ICA) or factor analysis (FA) here. These techniques suffer from the presence of negative entries in either the basis or the weight matrix or both. In the applications considered here, we assume non-negative observations to be generated by strictly additive superposition of non-negative basis vectors and there are no such negative entries.
Concluding this introductory chapter for NMF, we discuss its relation to another popular data analysis technique named *Probabilistic Latent Semantic Analysis* (PLSA).
Thomas Hofmann introduced PLSA [Hof99],[Hof01] for the analysis of binary and count data, which appear e.g. in text analysis. The technique performs a probabilistic mixture decomposition aimed to detect semantic features called aspects in documents. In this aspect model for co-occurrence data of words and documents, an observation is the appearance of word j in a particular document i. A generative model is defined by the following scheme:

1. Select a document d_i with probability $P(d_i)$
2. Pick a latent class z_k with probability $P(z_k|d_i)$
3. Generate a word w_j with probability $P(w_j|z_k)$

This corresponds to a joint probability model for words and documents

$$P(d_i, w_j) = P(d_i)P(w_j|d_i) = P(d_i) \sum_{k=1}^{K} P(w_j|z_k)P(z_k|d_i) \tag{2.51}$$

in which the conditional distributions $P(w_j|d_i)$ are convex combinations of the K aspects $P(w_j|z_k)$ such that $\sum_k P(z_k|d_i) = 1$.
The parameters can be learned in a maximum likelihood framework

$$\begin{aligned} LL: &= \sum_i \sum_j n(d_i, w_j) \ln P(d_i, w_j) \\ &= \sum_i n(d_i) \left[\ln P(d_i) + \sum_j \frac{n(d_i, w_j)}{n(d_i)} \ln \sum_k P(w_j|z_k)P(z_k|d_i) \right] \end{aligned} \tag{2.52}$$

where $n(d_i, w_j)$ is the number of occurrences of word j in document i and $n(d_i) = \sum_j n(d_i, w_j)$ is the length of document i. The log likelihood LL in eq. (2.52) can be maximized by the EM algorithm

[DLR77] which is a popular optimization technique for latent variable models. It alternates between an expectation (E) step where the posterior probabilities of the latent variables z_k are computed assuming the current estimates of the parameters and a maximization (M) step which updates the parameters based on the posterior probabilities estimated in the previous E-step.
The E-step for the PLSA model is given by

$$P(z_k|d_i, w_j) = \frac{P(w_j|z_k)P(z_k|d_i)}{\sum_k P(w_j|z_k)P(z_k|d_i)} \tag{2.53}$$

while in the M-step, the log-likelihood (2.52) has to be maximized w.r.t the normalization constraints $\sum_j P(w_j|z_k) = 1$ and $\sum_k P(z_k|d_i) = 1$. This yields the update rules

$$P(w_j|z_k) = \frac{\sum_i n(d_i, w_j)P(z_k|d_i, w_j)}{\sum_m \sum_i n(d_i, w_m)P(z_k|d_i, w_m)} \tag{2.54}$$

$$P(z_k|d_i) = \frac{\sum_j n(d_i, w_j)P(z_k|d_i, w_j)}{n(d_i)} \tag{2.55}$$

Shashanka et al. [SRS08] pointed out that the above EM equations can be written as

$$H_{kj} \leftarrow H_{kj} \sum_i \frac{X_{ij}}{[WH]_{ij}} W_{ik} \tag{2.56}$$

$$H_{kj} \leftarrow \frac{H_{kj}}{\sum_j H_{kj}} \tag{2.57}$$

$$W_{ik} \leftarrow W_{ik} \sum_j H_{kj} \frac{X_{ij}}{[WH]_{ij}} \tag{2.58}$$

using

1. $X_{ij} = n(d_i, w_j)$
2. $H_{kj} = P(w_j|z_k)$
3. $G_{ik} = P(d_i|z_k)$
4. a diagonal matrix $\mathbf{S}$ such that $S_{kk} = P(z_k)$, and
5. $\mathbf{W} = \mathbf{GS}$

and showed that PLSA is equivalent to the multiplicative Lee-Seung [LS99] algorithm for the KL-divergence measure which was discussed in eq. (2.7).
This relationship was also mentioned by others [GG05], [DLP08].
We discussed the relationship between NMF and PLSA in detail, since both techniques use algorithms which are numerically equivalent although being conceptually different. While NMF is usually formulated as constrained optimization problem involving a cost function, PLSA is a probabilistic model whose parameters can be estimated by the well-known EM-procedure [DLR77].
We will have a closer look on the statistical aspects of NMF later in section (7.1).

2.6 Summary

In this chapter the method Non-negative matrix factorization was introduced as a technique to decompose non-negative data into potentially meaningful components.

The extracted basis and weight matrices are both constrained to be non-negative, too.
We briefly reviewed the existing literature on NMF and discussed the basic challenges in the design of NMF algorithms.
Concluding this chapter, we note that there are two important aspects of NMF which remain partly unsolved questions and need further investigation:

1. Uniqueness
   Is there a principled way to determine an optimal solution of a given NMF problem?

2. Model order
   Is there a way to automatically determine an optimal number of components in a given NMF problem?

While a solution to the first question will be the topic of the next chapter, Bayesian approaches to answer both questions will be discussed in chapters (6) and (8).

Chapter 3

Uniqueness of NMF and the Determinant Criterion

In this chapter, we have a closer look on the problem concerning the uniqueness of NMF.
We propose a determinant criterion to constrain the solutions of non-negative matrix factorization problems and achieve unique and optimal solutions in a general setting, provided an exact solution exists. We demonstrate how optimal solutions are obtained by a heuristic named *detNMF* in an illustrative example and discuss the difference to sparsity constraints.

3.1 Uniqueness of NMF

NMF has seen numerous applications in recent years (see paragraph 2.2) where it has been primarily applied in an unsupervised setting for example in image and natural language processing, sparse coding, and a variety of applications in computational biology. Usually, NMF is used to gather hidden information from data, such as the extraction of (at least some of) the underlying components which can be used to generate the whole dataset by non-negative superposition.

3.1.1 Why is uniqueness important?

To make NMF a reliable data analysis tool, it is essential to understand its intrinsic insufficiency of a unique solution. Figure (3.1) illustrates two different problems concerning the optimal solutions of an optimization problem. In case A (left hand side) several points of the search space (or parameter space) lead to an equally low cost function. These minima are equivalent, since they can not be distinguished by evaluating the cost function. The related uniqueness problem (i.e. to assign one *best* solution among several equivalent ones) is an intrinsic problem of the system.
In the second case B displayed on the right hand side of figure (3.1), there is one global minimum and at least one local minimum. A downhill algorithm such as gradient descent can not traverse a minor bump of the cost function and become stuck into a local minimum, although there is another minimum with a lower cost. This problem can be circumvented by several runs using different starting positions, or stochastic search algorithms.
As we will see, the uniqueness problems of NMF are of the kind A: without additional constraints there are several equivalent solutions which can not be distinguished by a cost function which measures the reconstruction error only.
The existence of several solutions can lead to multiple interpretations. Running an identical analysis twice must not lead to different solutions without understanding their origin or judging their quality.

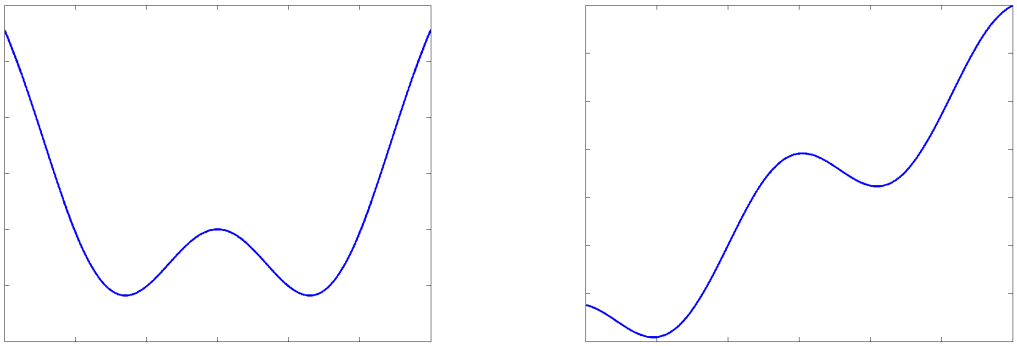

Figure 3.1: The x-axes denote the space of possible parameter settings, while the y-axes show the value of a cost function. In case A (left) there are several different optimal solutions which have an equal cost function. In contrast, case B has one global minimum and one or more local optima.

Established analysis tools like for example PCA have a fixed hierarchy of the extracted principle components, which are sorted according to their explained variance and orthogonal to each other.
NMF does not have such an ordering of the extracted components.
In contrast, there can be several equivalent solutions at an equal level of the cost function (type A).
In addition, NMF algorithms are greedy algorithms and can, on principle, only converge (if at all) to local minima of the cost function.
There are some recent attempts to deal with the non-uniqueness NMF-solutions. [DS04], poses the two fundamental questions

1. Under what assumptions is NMF well-defined, e.g. in some sense unique?

2. Under what assumptions is the factorization correct, recovering the right answer?

The idea of a simplical cone in the non-negative orthant spanned by the basis vectors which contains the data cloud is discussed, and situations are given (e.g. an ice-cream cone) where there is only one possible solution.
Other approaches add penalty terms to the plain cost function, such as the sparsity constraint on the weights used in non-negative sparse coding by Hoyer [Hoy02]

$$D_{nnsc} = \sum_i \sum_j (X_{ij} - [\mathbf{WH}]_{ij})^2 + \lambda \sum_{ik} W_{ik} \tag{3.1}$$

or the constraints used in the local NMF algorithm (LNMF) by Li et al. [LHZC03] encouraging spatially localized representations

$$D_{lnmf} = \sum_i \sum_j (X_{ij} \ln \frac{X_{ij}}{[\mathbf{WH}]_{ij}} - X_{ij} + [\mathbf{WH}]_{ij}) + \alpha \sum_{ia} [\mathbf{W}^T\mathbf{W}]_{ia} - \beta \sum_k [\mathbf{HH}^T]_{kk} \tag{3.2}$$

Various regularizing constraints have also been added to enforce certain characteristics of the solutions or to impose prior knowledge about the application considered [Hoy04]. Further, [TST05] analyzed the uniqueness properties of sparse NMF and prove that Hoyer's algorithm [Hoy04] indeed finds the closest points of fixed sparseness.

The development of positive matrix factorization [PT94], as surveyed in [Hop00], rather proposes application-driven extra penalties whose justification requires background knowledge.
In addition to the plain NMF cost function describing the reconstruction error, additional penalty terms for either $\mathbf{W}$ or $\mathbf{H}$ or both are added. The new constrained cost function then allows to distinguish between two solutions with equal reconstruction error by the penalty terms (e.g. two solutions which are equivalent according to the plain cost function become distinguishable due to some extra constraints). However, the choice of the penalty function and the determination of an appropriate balance between approximation error and penalty (e.g. the determination of an optimal λ in eq. (3.1) can be tricky and must be justified by additional knowledge.
Laurberg [LCP+08] constructed a restricted uniqueness theorem for NMF based on Plumbley's results on non-negative ICA [Plu01], [Plu02], [Plu03]. The circumstances under which NMF of an observed nonnegative matrix is unique are investigated and necessary and sufficient conditions for the uniqueness are given.
Summarizing, uniqueness of NMF is either enforced by *directly* imposing additional constraints, or conditions on the data are discussed under which unique solutions exist.
Here, we[1] suggest a different approach to uniqueness. Starting from a geometrical point of view similar to [SDB+03], we illustrate the problem of uniqueness in terms of the volume spanned by the basis vectors. We further explain how the determinant can be used to identify an optimal solution arising naturally among all possible exact non-negative decompositions of a given data matrix given no additional constraints.

3.1.2 Uniqueness

Let $\mathbf{X} = \mathbf{WH}$ be a non-negative factorization into K basis components. Obviously, these matrices $(\mathbf{W}, \mathbf{H})$ are far from representing unique solutions. Any pair $(\mathbf{WS}^{-1}, \mathbf{SH})$ still provides a valid decomposition of the same data $\mathbf{X}$ as long as $\mathbf{S}^{-1}$ exists and $\mathbf{WS}^{-1}, \mathbf{SH} \geq 0$. A unique solution for a given NMF problem can be specified by an arbitrary solution $(\mathbf{W}, \mathbf{H})$ together with a transformation $\mathbf{S}$ appropriately restricted by suitable non-negativity constraints.
Among all possible such transformations $\mathbf{S}$, there are two special cases :

1. $\mathbf{S}$ is a diagonal matrix with positive entries
2. $\mathbf{S}$ is a permutation matrix.

Case (1) represents a scaling invariance which can be overcome by normalization of the corresponding basis or feature vectors. For example, simultaneous scaling of a row $\mathbf{H}_{k*}$ by a scalar $\alpha_k > 0$ and the corresponding column $\mathbf{W}_{*k}$ by $\frac{1}{\alpha_k}$ does not alter their scalar product. We can then alleviate the scaling invariance e.g. by normalization of the feature vectors $\mathbf{H}_{k*}$. Sorting with respect to the norm of $\mathbf{W}$ then also removes the permutation freedom (2).
In general, there are much more than these two trivial indeterminacies for NMF, which we will discuss next.

3.2 Geometrical Approach

3.2.1 Problem illustration

Although real world observations are usually fraud with noise, we will restrict ourselves to the noise-free case in the following.

[1] originally proposed by G. Pöppel 2007, private communication

Of course, noise induces further uncertainty. However, the non-uniqueness problem of NMF we consider here has nothing to do with additional noise but is a basic property of the model formulation.
In the following, we assume the M-dimensional data $\mathbf{X}$ to be generated by non-negative linear combinations of $K \leq M$ linear independent feature vectors $\mathbf{H}_{k*} \geq 0$ such that $\mathbf{X} = \mathbf{WH}$ exactly.
We further assume the $\mathbf{H}_{k*}$ to be normalized, so that $\sum_j (H_{kj})^2 = 1$ for $k = 1, \ldots, K$ to fix the scaling of both $\mathbf{W}$ and $\mathbf{H}$.
The goal of any NMF-algorithm is the identification of feature vectors $\mathbf{H}_{k*}$, given only the data $\mathbf{X}$ and the number of features K.

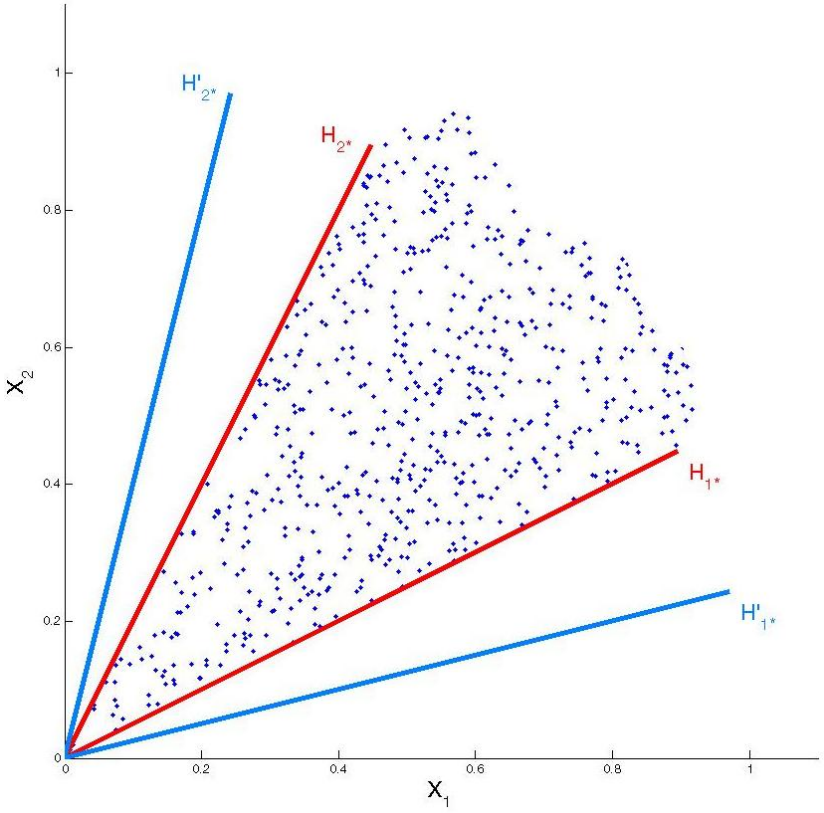

Figure 3.2: 2D illustration for NMF: Detail from data generated via $\mathbf{X} = \mathbf{WH}$, using a equally distributed 1000×2 matrix $\mathbf{W}$ in $[0, 1]$ and $\mathbf{H} = \sqrt{\frac{4}{5}} \left(\begin{smallmatrix} 0.5 & 1 \\ 1 & 0.5 \end{smallmatrix}\right)$. Two possible NMF-bases $(\mathbf{H}_{1*}, \mathbf{H}_{2*})$ and $(\mathbf{H}'_{1*}, \mathbf{H}'_{2*})$ are shown.

Examining all information (a-d) carefully

$$(a) \quad \mathbf{X} = \mathbf{WH}, \quad (b) \quad \mathbf{X} \geq 0, \quad (c) \quad \mathbf{W} \geq 0, \quad (d) \quad \mathbf{H} \geq 0 \tag{3.3}$$

leads to the following geometrical interpretation:

- $\mathbf{X} \geq 0$:
  All data vectors $\{\mathbf{X}_{i*}\}_{i=1}^N$ constitute a cloud in the non-negative orthant $(\mathbb{R}_0^+)^M$ of $\mathbb{R}^M$.

- $\mathbf{H} \geq 0$:
  Each basis vector $\mathbf{H}_{k*}$ points into $(\mathbb{R}_0^+)^M$; Normalized vectors $\mathbf{H}_{k*}$ span a K-dimensional subspace of the non-negative orthant $(\mathbb{R}_0^+)^M$ of $\mathbb{R}^M$.

- $\mathbf{W} \geq 0$:
  Positive coefficients W_{ij} for all j imply that a data vector $\mathbf{X}_{i*}$ is inside this K-dimensional subspace, whereas $W_{ij} = 0$ for at least one j indicates that point X_{i*} lies on a K-1 dimensional peripheral surface. Here, we consider the surfaces as part of the interior of the subspace.

(see Fig. 3.2 for a two-dimensional example where $M = K = 2$, see also [Hop00],[SDB+03]). Any set of K non-negative vectors $\{\mathbf{H}_{k*}\}_{k=1}^K$ which encloses the data $\mathbf{X}$ in a non-negative fashion provides a valid solution to the NMF-problem $\mathbf{X} = \mathbf{WH}$. Now let us recall the additional transformations $\mathbf{S}$ from paragraph 3.1.2 which converts the matrix $\mathbf{H} \mapsto \mathbf{SH}$ and $\mathbf{W}$ accordingly. Given a solution, $(\mathbf{W}, \mathbf{H})$, the resulting vectors $\mathbf{SH}_{k*}$ provide another solution, if all $\mathbf{SH}_{k*} \geq 0$ **and** all data lie inside the new parallelepiped ($\mathbf{WS}^{-1} \geq 0$).
A natural choice of a unique solution given the data $\mathbf{X}$ is a set of nonnegative vectors $\{\mathbf{H}_{k*}\}_{k=1}^K$, which span a parallelepiped of minimal volume containing $\mathbf{X}$. In other words, the non-negative K-tuple of M-dimensional vectors enclosing the data in the tightest possible way is our candidate for a unique solution. (See the pair $(\mathbf{H}_{1*}, \mathbf{H}_{2*})$ in Fig. 3.3, left.)

3.2.2 The Determinant Criterion

Let $P(\mathbf{H})$ be the parallelepiped spanned by the vectors $\mathbf{H}_{1*}, \ldots \mathbf{H}_{K*}$. If $\mathbf{H}$ is a square matrix, the volume of $P(\mathbf{H})$ is given by

$$vol(P) = |\det(\mathbf{H})|. \tag{3.4}$$

In case of a non-square $K \times M$- matrix $\mathbf{H}$, the volume of the parallelepiped can be obtained by the following relation: $vol(P) = \sqrt{det(\mathbf{HH}^T)}$. If we normalize each feature $\mathbf{H}_{k*}$ by $\left(\sqrt{\sum_j (H_{kj})^2}\right)^{-1}$, a minimal determinant relates to minimal angles between the edges of $P(\mathbf{H})$.
Thus, our candidate for an optimal solution can be specified as follows:

$$\mathbf{X} = \mathbf{WH} \tag{3.5}$$

normalized such that

$$\sqrt{\sum_j (H_{kj})^2} = 1, \text{ for } k = 1, \ldots, K \tag{3.6}$$

and

$$\det(\mathbf{HH}^T) = \min, \tag{3.7}$$

regarding $\mathbf{X}, \mathbf{W}$, and $\mathbf{H} \geq 0$.

3.2.3 Limitations of the determinant criterion

Assuming a noise-free model, there is at least one exact solution for the correct K. The optimal solution is given by the one selected by the determinant criterion (eqns. 3.5, 3.5, 3.5), and is unique up to permutation.
There are two important special cases, where the determinant criterion fails to describe a unique optimal solution:

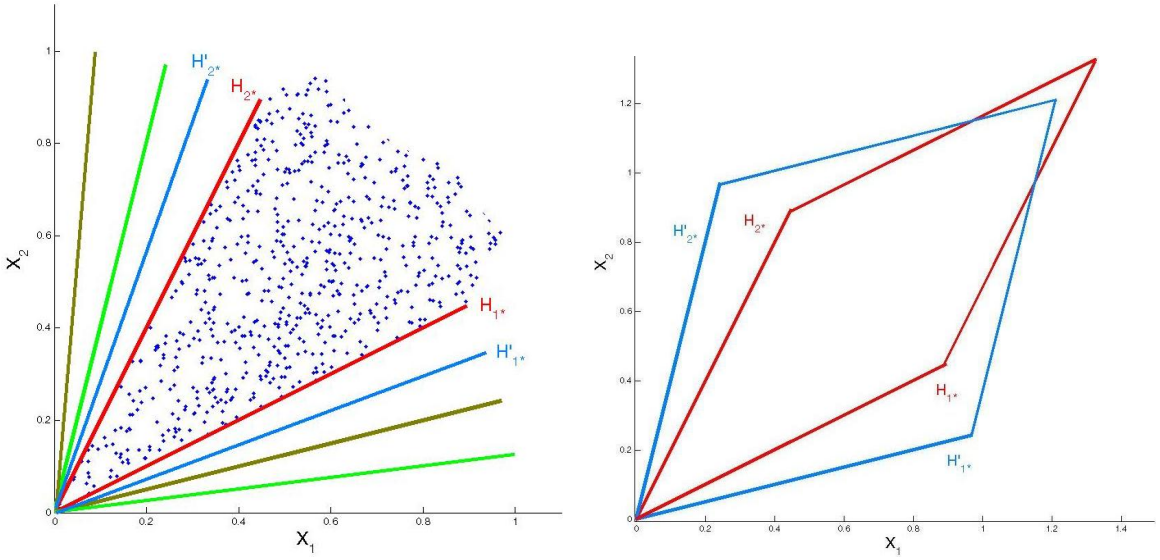

Figure 3.3: *left*: Non-uniqueness geometrically: there are several possible sets of basis vectors $\mathbf{H} \geq 0$ which enclose the given data $\mathbf{X} \geq 0$ in a non-negative fashion $\mathbf{W} \geq 0$. *right*: If the basis vectors $\mathbf{H}_{k*}$ are normalized to unit length, the volume of the spanned parallelepiped is directly related to the angle at the origin.

1. Rotational symmetry (see Fig. 3.4, left)
   In case there is rotational symmetry in the data, several equivalent sets of basis vectors can span an identical volume. In the extremal case, the data describes a cone and there are infinite possibilities to enclose this. If there is rotational symmetry, there are several equivalent optimal solutions, which cannot be distinguished by the determinant criterion. Note that the data need not be completely rotational symmetric as in the example in figure 3.4, left. For example, data with a polygonal cross-section can also lead to rotational ambiguity.

2. Offset (see Fig. 3.4, left)
   If every data point shares a fixed contribution vector $\mathbf{H}_{0*}$, the solution of minimal volume is not identical to the optimal one. This special case is known as affine NMF problem and is discussed in [LH07]. The offset can be incorporated explicitly to the NMF problem.

If the data is noisy and the model $\mathbf{X} \approx \mathbf{WH}$ holds only approximately, a suitable trade off between the reconstruction accuracy and a minimal determinant has to be found. In that case, the existence of unique solutions is arguable anyway.

3.2.4 The Algorithm *detNMF*

In this paragraph we discuss a new algorithm named *detNMF*, which directly implements the determinant criterion. As an objective function, we chose the regularized least-squares function

$$E(\mathbf{X}, \mathbf{WH}) = \sum_{i=1}^{N} \sum_{j=1}^{M} (X_{ij} - [\mathbf{WH}]_{ij})^2 + \alpha \det(\mathbf{HH}^T), \tag{3.8}$$

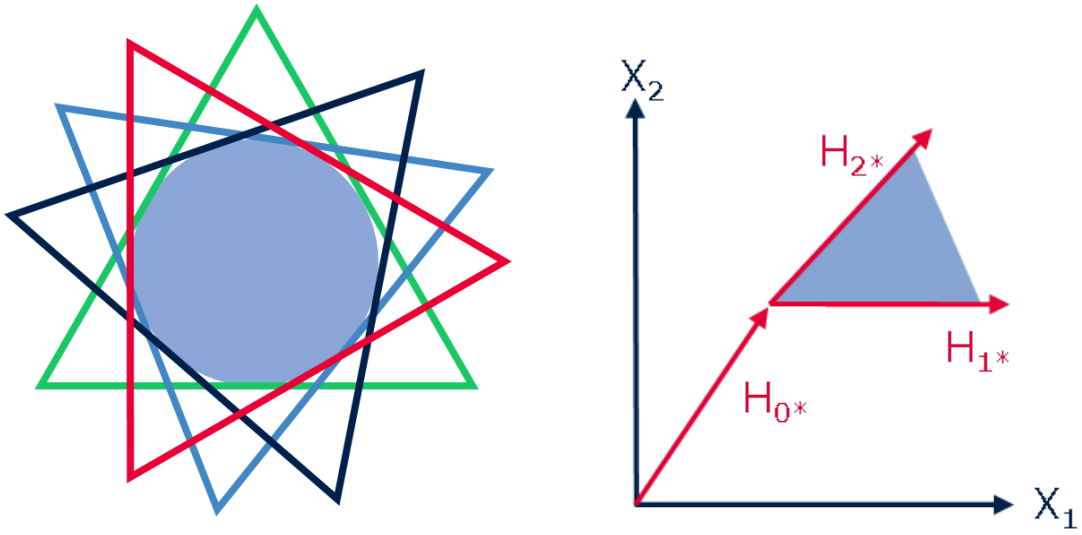

Figure 3.4: Cases where the determinant criterion does not identify the correct solutions *left*: The data is rotational symmetric. The origin lies behind plane of projection. Each triangle describes a different triple of equivalent enclosing basis vectors $\mathbf{H}_{1*}$, $\mathbf{H}_{2*}$, $\mathbf{H}_{3*}$. *right*: The data has an offset. Each data vector shares an offset $\mathbf{H}_{0*}$ such that the enclosing basis vectors do not intersect in the origin. (affine NMF [LH07])

where α is a small positive regularization parameter. The regularizing term $\det(\mathbf{H}\mathbf{H}^T)$ is differentiable, and its partial derivatives are given by

$$\frac{\partial \det(\mathbf{H}\mathbf{H}^T)}{\partial H_{lm}} = 2\det(\mathbf{H}\mathbf{H}^T)[(\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{H}]_{lm}. \tag{3.9}$$

Now consider the following multiplicative update rules (3.10)-(3.13 within the algorithm *detNMF* : update:

$$H_{lm} \leftarrow H_{lm} \cdot \frac{[\mathbf{W}^T\mathbf{X}]_{lm} - \alpha \det(\mathbf{H}\mathbf{H}^T)[(\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{H}]_{lm}}{[\mathbf{W}^T\mathbf{W}\mathbf{H}]_{lm}} \tag{3.10}$$

$$W_{lm} \leftarrow W_{lm} \cdot \frac{[\mathbf{X}\mathbf{H}^T]_{lm}}{[\mathbf{W}\mathbf{H}\mathbf{H}^T]_{lm}} \tag{3.11}$$

normalize:

$$H_{lm} \leftarrow \frac{H_{lm}}{\sqrt{\sum_m (H_{lm})^2}} \tag{3.12}$$

$$W_{ml} \leftarrow W_{ml} \cdot \sqrt{\sum_m (H_{lm})^2} \tag{3.13}$$

These update rules represent a modification of the well-known Lee and Seung method [LS99], [LS01] to which they reduce in the limit $\alpha \to 0$ (compare eq. 2.6 in paragraph 2.3.2). General methods of how to regularize multiplicative updates with additional constraints can be found in [BBL+06] or [DS05].

3.2.5 The choice of α

If the parameter α in the reformulated update rule (3.10)

$$H_{lm} \leftarrow H_{lm} \left(\frac{[\mathbf{W}^T\mathbf{X}]_{lm}}{[\mathbf{W}^T\mathbf{W}\mathbf{H}]_{lm}} - \alpha \frac{\det(\mathbf{H}\mathbf{H}^T)[(\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{H}]_{lm}}{[\mathbf{W}^T\mathbf{W}\mathbf{H}]_{lm}} \right) \tag{3.14}$$

is set to zero, the update rule reduces to the Lee and Seung multiplicative update rule. The latter has been shown not to increase the reconstruction error [LS01], [Lin07a], [SD06].
As described in Section 3.2, any solution to an exactly solvable NMF-problem requires the basis vectors $\mathbf{H}_{k*}$ to lie on the periphery of the data cloud. In order to minimize the reconstruction error under the non-negativity constraint, a NMF algorithm thus tries to position the basis vectors $\mathbf{H}_{k*}$ outside the given data cloud. In contrast, the determinant constraint forces the basis vectors $\mathbf{H}_{k*}$ to move towards each other for $\alpha > 0$, which contradicts the requirement that the basis vectors have to include all data points into their convex hull. Hence, α should be kept small or zero during the first iterations and increased as soon as an outer position of the basis vectors is reached. Since the algorithm is initialized by random matrices $\mathbf{W}$ and $\mathbf{H}$, a general rule for the choice of the parameter α is hard to find. In simulations we observed that if α is always kept small enough so that the reconstruction error does not increase during an iteration, very satisfactory results are obtained with the above algorithm. This strategy does not spoil the approximate, constrained decomposition of the algorithm as it only pushes the basis vectors towards the cloud of data points if this is possible.

3.3 Illustrative example

In this section the determinant criterion is illustrated by simulations. First, we show that plain NMF without additional constraints produces various distinct decompositions of an artificial data set, whereas the *detNMF* algorithm consistently recovers the correct solution in multiple runs.

3.3.1 Unconstrained NMF versus *detNMF*

The following simulation setup was used. A fixed 3×5 matrix $\mathbf{H}$ representing basis components of characteristic shapes (see also Fig. 3.5, left)

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_{1*} \\ \mathbf{H}_{2*} \\ \mathbf{H}_{3*} \end{pmatrix} = \begin{pmatrix} \textit{bowl} \\ \textit{stair} \\ \textit{block} \end{pmatrix} \tag{3.15}$$

The true weight matrix $\mathbf{W}$ is generated by non-negative equally distributed random numbers in $[0, 1]$. The algorithm receives $\mathbf{X} = \mathbf{W}\mathbf{H}$ and the correct factorization rank $K = 3$ as input. The matrices $\mathbf{H}$ and $\mathbf{W}$ are initialized with random positive numbers and then iteratively updated as described above until the normalized reconstruction error falls below a certain threshold (e.g. $\frac{E(\alpha=0)}{NM} < 10^{-10}$).
Figure 3.7 shows a comparison of the results gained via plain (unconstrained) NMF and *detNMF*. The pictures show 3-dimensional projections of 10 randomly initialized runs using equally and sparsely distributed weights $\mathbf{W}$ and $\mathbf{H}$ as in Figure 3.5. While the *detNMF* algorithm consistently recovers the original basis vectors $\mathbf{H}$ which constitute the edges of a tetrahedron in a three dimensional projection, the solutions gained via plain NMF without additional constraints vary.

In the simulations, the algorithm *detNMF* always extracted the correct features despite starting with random initializations as well as different original coefficient matrices $\mathbf{W}$ and varying numbers of individuals (e.g. $N = 100, 1000, 10000$).

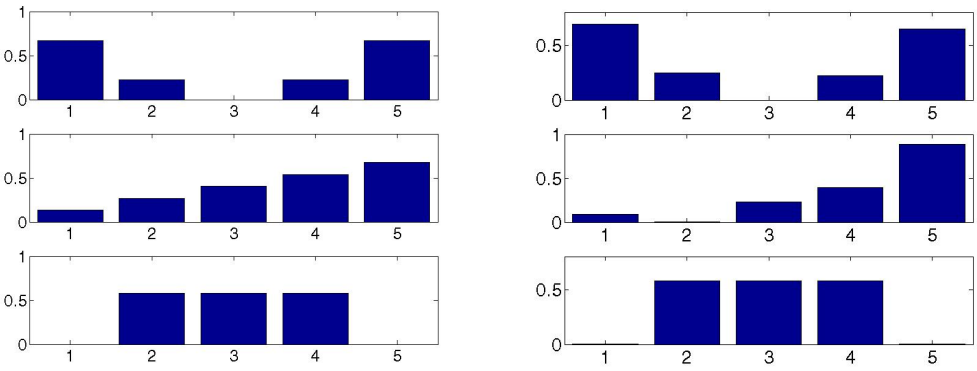

Figure 3.5: *Left*: Original features *bowl* $\mathbf{H}_{1*}$(top), *stair* $\mathbf{H}_{2*}$(center), and *block* $\mathbf{H}_{3*}$(bottom), which are perfectly recovered by the *detNMF* algorithm. Each vector $\mathbf{H}_{k*}$ is a 5-dim analogon to the 2-dim vectors in Fig. 3.2. *Right*: Example of a valid solution, but with wrong features $\mathbf{H'}_{1*}, \mathbf{H'}_{2*}, \mathbf{H'}_{3*}$, obtained via unconstrained NMF

During iteratively reconstructing the data matrix $\mathbf{X}$ with sufficient precision, the determinant criterion pushes the feature vectors towards the optimal solution with the smallest possible determinant (see Fig. 3.6). In contrast, the unconstrained version ($\alpha = 0$, as proposed by [LS99]) converges to several different solutions, depending on the initialization of the NMF procedure. In Fig. 3.5 (right) we give an example of an exact nonnegative factorization of the data in $(\mathbf{W'}, \mathbf{H'})$ which does not reproduce the original features $\mathbf{H}$ correctly. Note that $\mathbf{H'}_{2*} + 0.2\mathbf{H'}_{3*} \approx \mathbf{H}_{2*}$. Furthermore, note that $\det(\mathbf{H}\mathbf{H}^T) = 0.18$ and $\det(\mathbf{H'}\mathbf{H'}^T) = 0.31$. The basis $\mathbf{H'}$ is sufficient to explain all data by a non-negative superposition, but does not coincide with the correct solution which generated the data and is characterized by a minimal determinant.

3.3.2 Determinant Criterion versus Sparseness Constraints

In the following, we will demonstrate that the determinant criterion offers a natural way to induce unique solutions to exactly solvable NMF problems, whereas sparseness constraints as suggested by [Hoy02] or [LHZC03] among others can be misleading. We discuss two extreme toy examples:

1. In the first example our *detNMF* algorithm correctly discovers the unique features of a non-sparse data distribution, whereas a sparse NMF approach fails to do so.

2. In the second example we use a data distribution with a multi-modal density which lends itself to a very sparse representation.

With these extreme examples we intend to highlight the fact that sparse coding does not represent a robust and natural criterion for unique solutions, while a determinant criterion can be, provided enough data is available.
For comparison, we chose the *nnsc*-algorithm, as described in [Hoy02] (*nnsc*: non-negative sparse coding), which is tailored to provide good solutions for sparse data sets. It minimizes the following objective function

$$E_{nnsc} = \sum_{i=1}^{N}\sum_{j=1}^{M}(X_{ij} - [\mathbf{WH}]_{ij})^2 + \lambda \sum_{ij} W_{ij}. \tag{3.16}$$

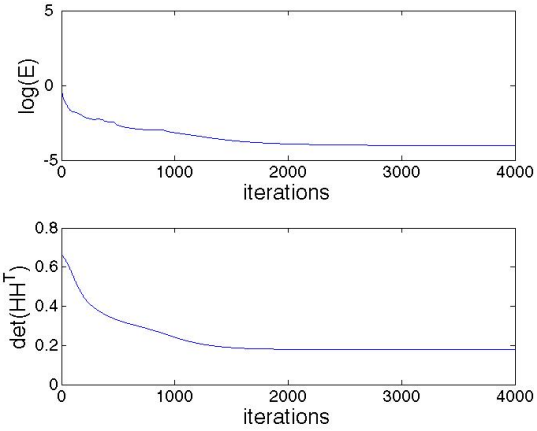

Figure 3.6: Typical evolution of simulation parameters using the determinant criterion. *top*: Logarithm of the reconstruction error, *bottom*: determinant $\det(\mathbf{H}\mathbf{H}^T)$

The term $\lambda \sum_{ij} W_{ij}, \lambda \geq 0$ penalizes large mixing coefficients W_{ij}, hence pushes the solution towards small coefficients. Ideally, using an appropriate sparsity constraint, most data points should be explained by a minimal set of basis vectors $\mathbf{H}_{s*}$, with the majority of all other coefficients $W_{ij}, j \neq s$ being zero. An optimally sparse $\mathbf{W}$ thus corresponds to a solution where the columns $\mathbf{W}_{*j}$ contain as much zeros as possible.
With the first example we discuss a data distribution which is concentrated right in the middle between the basis vectors which span the data space. This non-sparse data distribution serves to demonstrate potential drawbacks of sparsity constraints by means of the following idea:
The representation of all data points using only non-negative expansion coefficients, i.e. ($\mathbf{W} \geq \mathbf{0}$), requires all basis vectors to lie on the periphery of the data subspace (see Fig. 3.2). If, however, the NMF algorithm tries to satisfy a sparseness condition by approaching one basis vector to a data agglomeration which is not located near a true basis vector, the non-negativity constraint instantly forces other basis vectors to balance this close approach by moving further away from the cloud of data points. Otherwise, not all non-negativity constraints can be met. The property of maximal sparseness, imposed by some suitable criterion, thus can lead to a valid unique solution which might not be the solution searched for and certainly does not represent a minimum volume solution. On the other hand we expect our minimal volume constraint to cope with the situation and to yield the correct solution.
Example 2 demonstrates that even in cases of very sparse data distributions where any sparse NMF algorithm is expected to yield good results our minimal volume constraint also should yield unique and correct solutions.
In case of example 1 and using the feature vectors $\mathbf{H}_{k*}$ (eq. 3.15), we construct the coefficients in $\mathbf{W}$ as follows: 90% of the data points are generated via $s(t \cdot \mathbf{H}_{1*} + (1-t) \cdot \mathbf{H}_{3*})$, where the parameter t is randomly drawn from a Gaussian distribution with $(\mu, \sigma) = (0.5, 0.03)$, and s equally distributed in the interval $[0, 1]$. Being projected onto the first three principal components of $\mathbf{X}$, the feature vectors $\mathbf{H}_{k*}$ constitute the edges of a tetrahedron (see Fig. 3.8, solid lines). Note that by construction, the data has exactly three principal components related to nonzero eigenvalues. Most data points are an approximately equally weighted mixture of two features and thus lie on a surface between two edges of the tetrahedron. The remaining 10% of the data are equally distributed in the space between the feature vectors as illustrated in Fig. 3.8.

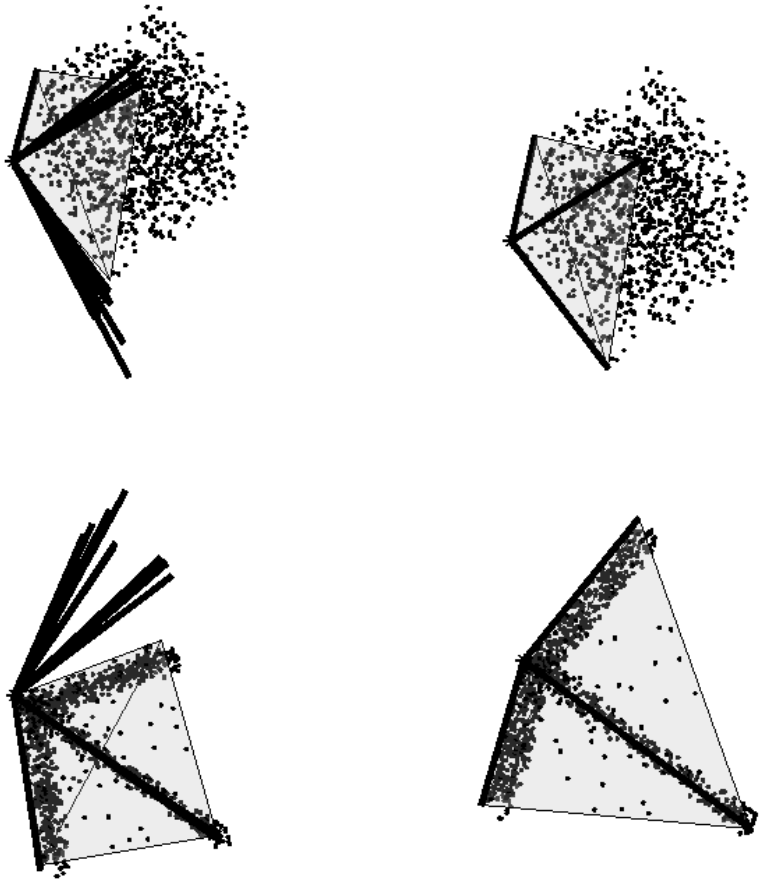

Figure 3.7: Comparison of plain NMF (left) and *detNMF* (right). The results of 10 randomly initialized runs are shown. *top*: **W** equally distributed; *bottom*: **W** sparse

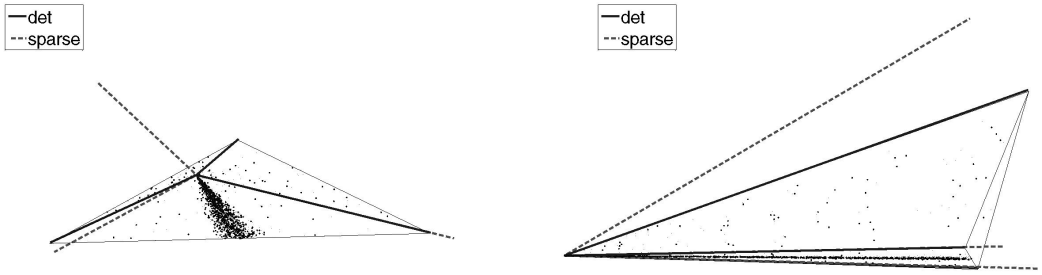

Figure 3.8: 3D-visualization of the data distribution of example 1 (see text for details). Data and feature vectors are projected onto the three principal components of the data. In this space, the original feature vectors $\mathbf{H}_{1*}$, $\mathbf{H}_{2*}$, $\mathbf{H}_{3*}$ constitute the edges of a tetrahedron with length 1. These features are exactly recovered by the *detNMF* algorithm (solid lines). Obviously, the sparse NMF approach fails to position all basis vectors correctly. Note that the normalized feature vectors deduced from the *nnsc* algorithm are drawn on a larger scale (dashed lines) to render them visible as two feature vectors, deduced with both algorithms, almost coincide. All feature vectors intersect at the vertex located at the origin. *left*: top view, the vertex at the origin is placed in the center of the figure *right*: side view, the left corner represents the origin of the axis system.

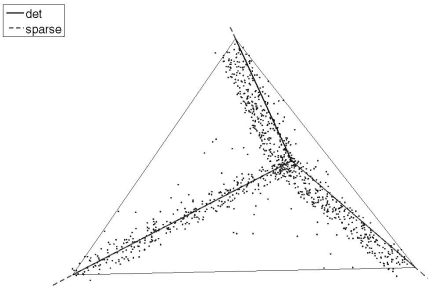

Figure 3.9: 3D-visualization of the data distribution of example 2 (see text for details). Data and feature vectors are projected onto the three principal components of the data. In this space, the original feature vectors $\mathbf{H}_{1*}$, $\mathbf{H}_{2*}$, $\mathbf{H}_{3*}$ constitute the edges of a tetrahedron with length 1. These features are recovered by both the *nnsc* algorithm (broken lines) and the *detNMF* algorithm (solid lines). All feature vectors intersect at the vertex located at the origin which is placed in the center of the figure.

After random initialization of $\mathbf{W}$ and $\mathbf{H}$, we processed both algorithms (*detNMF,nnsc*) until the reconstruction error ($(NM)^{-1}E(\alpha = 0)$ in eq. 3.8 and $(NM)^{-1}E_{nnsc}(\lambda = 0)$ in eq. 3.16, respectively) was smaller than 10^{-10}. Again, the *detNMF* algorithm recovered the correct feature vectors. On the other hand, the *nnsc* algorithm produced a solution with a smaller value of $\sum_{i,j} W_{ij}$, but also comprising a larger determinant than corresponds to the original data matrices. The final values of

	example 1		example 2	
	$\det(\mathbf{H}\mathbf{H}^T)$	$\sum_{i,j} W_{ij}$	$\det(\mathbf{H}\mathbf{H}^T)$	$\sum_{i,j} W_{ij}$
detNMF	0.1806	495.6798	0.1803	583.3226
nnsc	0.3096	487.7167	0.1804	583.2772

Table 3.1: Comparison of the values of the constraints for both algorithms, taken at the solutions gained by the algorithms *detNMF* and *nnsc*

the constraints are given in Tab. 3.1, while a $3D$-visualization of the data and the resulting feature vectors are shown in Fig. 3.8.
As is obvious from Table 3.1(left) and Figure 3.8, the complete set of all three features is not identified correctly if a maximally sparse solution is enforced. Though the majority of the data points is not accumulated near the feature vectors, a maximally sparse matrix $\mathbf{W}$ still exists which, however, does not represent the true matrix of mixing coefficients. While two feature vectors $\mathbf{H'}_{i*}$ are oriented such as to satisfy the sparseness constraint, a third feature vector moves far away from the given data points. This movement is enforced by the overall non-negativity constraint. Thus, if the data is not maximally sparse, the decomposition achieved by a sparse NMF-algorithm is not necessarily the correct one. We note that there are sparse algorithms (see e.g. [Hoy04], among others) containing an adjustable sparseness parameter which enables the selection of solutions containing a desired degree of spareness. However, such algorithms cannot lead to unique solutions in a general setting, since they require prior knowledge of the true sparseness of the matrix $\mathbf{W}$.
On the other hand, the determinant approach is more general, because it does not depend on the distribution of coefficients directly. It is based on the geometrical consideration that a true basis vector should only point into a certain region of space, if the data necessitates it because of the non-negativity constraints. The determinant criterion keeps the basis vectors away from such regions and hence constrains the solutions in a general way converging to optimal and unique solutions.
The second toy example is constructed in a way that a sparse NMF algorithm should be able to yield a good solution and identify the underlying features. Initializing the 1000×3- matrix $\mathbf{W}$ by equally distributed random numbers from the interval $[0, 1]$, in 90% of the triples (W_{i1}, W_{i2}, W_{i3}) two out of the three components were scaled by a factor of 10^{-1} in equal shares (see Fig. 3.9). In fact, the sparse algorithm did recover the correct features. But the minimal determinant constraint leads to an equally good solution as can be seen from Table 3.1.
Both examples demonstrate that the minimal determinant constraint leads naturally to unique and correct solutions independent of the characteristics of the given data distribution.

3.4 The multilayer technique

As mentioned in [CZA07], [CZA08], a sequential decomposition of nonnegative matrices as a cascade of $L > 1$ unmixing systems usually yields a better performance than one single decomposition.
According to [CZA07], it is an open theoretical issue to prove mathematically or explain more rigorously why the multilayer approach considerably improves the performance of NMF (or its tensor-extension NTF). The authors provide the following intuitive explanation: [...] *the multilayer system provides a sparse distributed representation of basis matrices* $\mathbf{H}^{(l)}$ *, so even a true basis matrix* $\mathbf{H}$ *is not sparse it can be represented by a product of sparse factors. In each layer we force (or encourage) a sparse representation. We found by extensive experiments that if the true basis matrix is sparse, most standard NTF/NMF algorithms have improved performance* [...] *. However, in practice not all data*

provides a sufficiently sparse representation, so the main idea is to model any data by cascade connections of sparse sub- systems. On the other hand, such multilayer systems are biologically motivated and plausible.

It turns out that the determinant criterion introduced above gives a simple alternative explanation for the success of multilayer techniques.

The multilayer technique can be explained as follows: In the first step, the decomposition $\mathbf{X} \approx \mathbf{W}^{(1)}\mathbf{H}^{(1)}$ is performed where $\mathbf{W}^{(1)}$ is a $N \times K$ and $\mathbf{H}^{(1)}$ is a $K \times M$ -matrix. The second step then considers the decomposition: $\mathbf{W}^{(1)} \approx \mathbf{W}^{(2)}\mathbf{H}^{(2)}$ (the weight matrix of the first step is interpreted as new data matrix and further decomposed). The l-th step then uses the result from the previous $l-1$ step and decomposes $\mathbf{W}^{(l-1)} \approx \mathbf{W}^{(l)}\mathbf{H}^{(l)}$ where $\mathbf{H}^{(l)}$ is a square $K \times K$ -matrix. Thus, a L-stage multilayer system performs the decomposition

$$\mathbf{X} \approx \mathbf{W}^{(L)}\mathbf{H}^{(L)}\mathbf{H}^{(L-1)} \dots \mathbf{H}^{(2)}\mathbf{H}^{(1)}. \tag{3.17}$$

Setting $\mathbf{H}^{(L)}\mathbf{H}^{(L-1)} \dots \mathbf{H}^{(2)}\mathbf{H}^{(1)} =: \mathbf{H}$, the determinant criterion reads

$$\begin{aligned} \det\left(\mathbf{H}\mathbf{H}^T\right) &= \det\left(\mathbf{H}^{(L)} \dots \mathbf{H}^{(2)}\mathbf{H}^{(1)}\mathbf{H}^{(1)T}\mathbf{H}^{(2)T} \dots \mathbf{H}^{(L)T}\right) \\ &= \det\left(\mathbf{H}^{(L)}\mathbf{H}^{(L)T} \dots \mathbf{H}^{(2)}\mathbf{H}^{(2)T}\mathbf{H}^{(1)}\mathbf{H}^{(1)T}\right) \\ &= \det\left(\mathbf{H}^{(L)}\mathbf{H}^{(L)T}\right) \dots \det\left(\mathbf{H}^{(2)}\mathbf{H}^{(2)T}\right) \det\left(\mathbf{H}^{(1)}\mathbf{H}^{(1)T}\right). \end{aligned} \tag{3.18}$$

Fixing the scaling freedom in every subproblem by normalizing the rows of $\mathbf{H}^{(l)}$ such that $(\sum_j (H^{(l)}_{kj})^2) = 1$ for all $k = 1, \dots, K$ implies that

$$0 \leq \det\left(\mathbf{H}^{(l)}\mathbf{H}^{(l)T}\right) \leq 1 \tag{3.19}$$

for every term $l = 1 \dots L$ in the last line of (3.18).

In a cascade of L sub-mixing steps, the the determinant $\det\left(\mathbf{H}\mathbf{H}^T\right)$ is a product of L terms, each of which is smaller or equal one. Thus, **the more substeps are performed, the more probable is the occurrence of at least one factor < 1 and thus a smaller overall determinant**.

As discussed above, the l-th subsystem $\mathbf{W}^{(l-1)} \approx \mathbf{W}^{(l)}\mathbf{H}^{(l)}$ can be interpreted as decomposition of the fictitious data $\mathbf{W}^{(l-1)}$ into coefficients $\mathbf{W}^{(l)}$ and a basis $\mathbf{H}^{(l)}$ and constitutes a usual NMF problem which can have several equivalent solutions. Any solution positions the basis vectors $\mathbf{H}^{(l)}_{k*}$ on the periphery of the data cloud $\mathbf{W}^{(l-1)}$. According to the determinant criterion the solution comprising the smallest volume is optimal for the particular subproblem.

The more demixing sub-steps are performed, the higher is the probability to obtain a solution with a small determinant. The multilayer technique is a fast and cheap option which produces minimum determinant solutions indirectly. In my experiments, usually 3-4 substeps are sufficient for consistent decompositions.

3.5 Related work

3.5.1 Endmember extraction

As an anonymous referee pointed out, a similar concept of a volume constraint for NMF has been independently developed to solve spectral unmixing problems [MQ07]. The idea of an enclosing simplex of minimum volume is known as *endmember extraction* hyperspectral unmixing applications [Cra94]. Although the idea is the same, it is limited to mixtures (where the mixing coefficients sum to one). Moreover, the approach by [MQ07] has a somewhat circumstantial implementation involving a PCA, which is not necessary. A recent study found both, Miao's [MQ07] and our approach [SPTL09] useful as an input for a Bayesian sampling procedure, noting that our approach is simpler [ASL09].

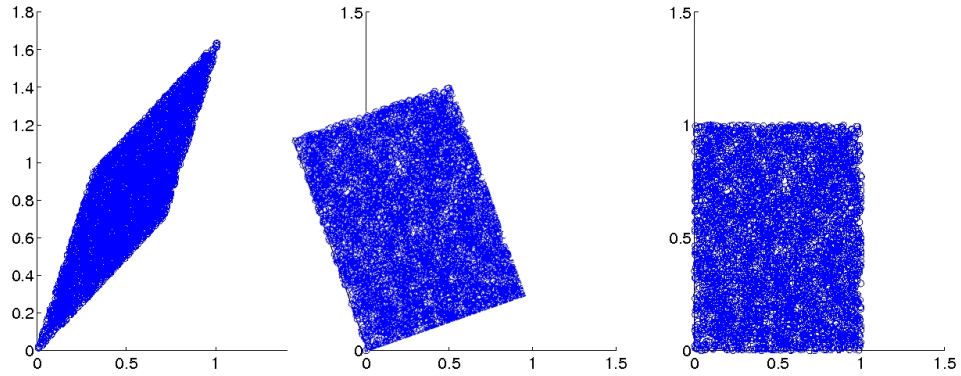

Figure 3.10: Non-negative ICA left: mixed signals; center: whitened signals; right: signals rotated to non-negative orthant

3.5.2 Non-negative ICA

Simultaneously with the development of NMF, Plumbley [Plu01], [Plu02], [Plu03] worked on a non-negative version of Independent Component Analysis (ICA) [Com94]. He stated that the non-negativity of the estimated sources together with the pre-whitening of the observed data is sufficient to recover the underlying non-negative sources uniquely.
While usual ICA algorithms determine the correct rotation after a whitening step by some non Gaussianity measure (e.g. the kurtosis), non-negative ICA utilizes the non-negativity constraint on the sources to determine this rotation. Thus, second-order decorrelation (instead of statistical independence) together with non-negativity constraints is sufficient to solve the non-negative ICA problem.
As depicted in figure 3.10, non-negative ICA first decorrelates the data and then rotates until the data fits into the non-negative orthant. In contrast, determinant constrained NMF does not need decorrelated features and directly discovers the basis vectors. Moreover, NMF is robust to additive noise (see e.g. [LCP+08]).

3.6 Conclusion

A determinant criterion was introduced to constrain the possible solutions of an exact NMF problem. Geometrically, this criterion means a minimum volume constraint on the subspace spanned by the basis vectors and emphasizes unique best solutions for a given problem. An easy to implement algorithm called *detNMF* which directly incorporates the determinant criterion was used in illustrative toy examples which represent two extreme data distributions. In these extremal settings, the *detNMF* algorithm was contrasted with a sparse NMF variant to demonstrate that sparseness constraints can be a misleading restriction while the determinant criterion is a more general approach. Moreover, the determinant criterion provides a very concrete explanation why a cascade of consecutive decompositions usually improves the performance of any NMF algorithm.

Chapter 4

NMF application to BIN data

4.1 Data aggregation

As explained in the introductory section about wafer fabrication (1.2.1), chips are usually tested in different BIN categories.

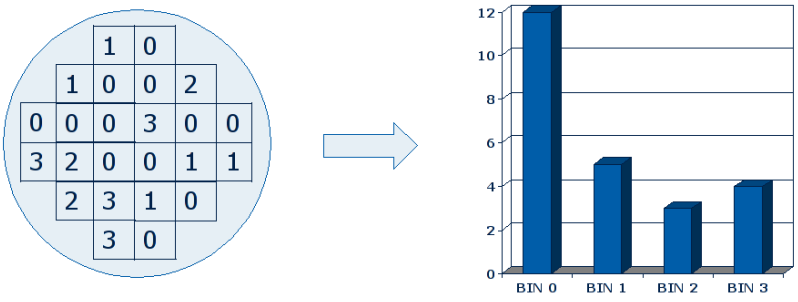

Figure 4.1: For the present investigation, each wafer is represented by (normalized) BIN counts. The example wafer (left) contains 24 chips, each labeled by one of the BINs $0, 1, 2, 3$. Here, BIN 0 means: *die works fine*, while BIN l (l=1,2,3) codes the event *chip failed in test category* l. The diagram on the right displays the same wafer in terms of BIN counts which can be further normalized by the total number of chips per wafer. If one ignores the pass BIN (BIN 0 here), the normalized BIN counts are an approximation of the fail probability in the respective BIN category.

We represent the i'th wafer by the M-dimensional row vector

$$\mathbf{X}_{i*} = (X_{i1}, \ldots, X_{iM}) \geq 0 \tag{4.1}$$

containing the number of chips carrying BIN label j, divided by the total number of chips per wafer on position X_{ij}. If the BIN number represents a test category where the chip can fail (usually any BIN category except the *pass BIN*), the matrix entry X_{ij} can be interpreted as an approximate probability that a chip on wafer i fails in category j.
We assume there is a set of $K \in \{2, 3, 4, \ldots\}$ underlying sources which are responsible for the fail chips. Each of the sources has an associated M-dimensional vector or typical fingerprint

$$\mathbf{H}_{k*} = (H_{k1}, \ldots, H_{kM}) \geq 0 \tag{4.2}$$

expressing the probability that a chip fails in category j due to source k in entry $H_{kj} \geq 0$.
We further assume that the fingerprint of each source $\mathbf{H}_{k*}$ remains the same, irrespective of the intensity of that source. The intensity of the k'th source on wafer i is assumed to be represented by the non-negative scalar $W_{ik} \geq 0$, where $W_{ik} = 0$ means that source k does not contribute to observation i and a value $W_{ik} > 0$ is a relative measure for the intensity.
In summary, we assume that an observation can be represented as linear combination

$$\mathbf{X}_{i*} \approx \sum_{k=1}^{K} W_{ik}\mathbf{H}_{k*} \tag{4.3}$$

of non-negative weights W_{ik} and basis components H_{k*}.
Data matrix entry $X_{ij} \geq 0$ contains the number of chips carrying BIN label j on wafer i, divided by the total number of chips on the wafer.
As a first approximation, the linear model given by eq. (4.3) holds true.
However, there are some nonlinear effects in the wafer test data, which we neglect here. One of these nonlinearities for example is induced by the fact that each chip carries only the label of one BIN category in this example, although if the chip can potentially fail in several different BIN categories. For example, if test A is performed before B, the apparent probability that a chip on wafer i fails in test B decreases if many chips fail in test A already since we associated an approximate fail probability with the number BIN counts per total chips.
Here, we assume that such nonlinear effects are small and interpret them as noise which is absorbed in the *approximately* in eq. (4.3).
In general, the linear approximation is quite accurate if the assumed fail probabilities are small.

4.2 Results

In this study, a data set comprising $N = 2800$ wafers and $M = 19$ BIN categories is used. A three-layered ALS NMF algorithm (see section 3.4) followed by refinement using the *detNMF* algorithm (see section 3.2.4) was applied to blindly recover the non-negative weight and basis matrices $\mathbf{W}$ and $\mathbf{H}$ from the $N \times M$ data matrix $\mathbf{X}$.
Note that unlike in PCA, the extracted basis vectors are not orthogonal to each other and are not extracted in a hierarchical order.
For NMF, we eliminate the scaling ambiguity by normalization such that each row vector has unit length ($\sum_j H_{kj}^2 = 1$). We sort the rows $\mathbf{W}_{*k}$ of the rescaled $\mathbf{W}$ in descending order by value of ($\sum_i W_{ik}^2$).
For illustration purposes, a PCA was performed and the data is projected onto the first three principal components (which have nothing to do with the NMF basis vectors!), see figure (4.2, left hand side) for a plot of the data. The majority is spread near the center at the origin, while some arms point outwards in a star-shaped manner. Note that the displayed 3 dimensional projection is used for visualization purposes only and cannot display the whole structure of the data.
On the right hand side of figure (4.2) the first five basis vectors extracted by NMF are displayed in addition. Each basis vector points into the direction one arms.

4.2.1 Examples for NMF decompositions

The NMF decompositions in different numbers of components K are quite consistent, as can be seen in figures (4.3) and (4.4). All basis vectors $\mathbf{H}^{K=3}$ corresponding to $K = 3$ components appear in $\mathbf{H}^{K=4}$ plus an additional component (see Fig. 4.3, top). The corresponding columns of the weight matrices $\mathbf{W}^{K=3}$ and $\mathbf{W}^{K=4}$ show the same behavior as indicated by the scatterplots given in (Fig.

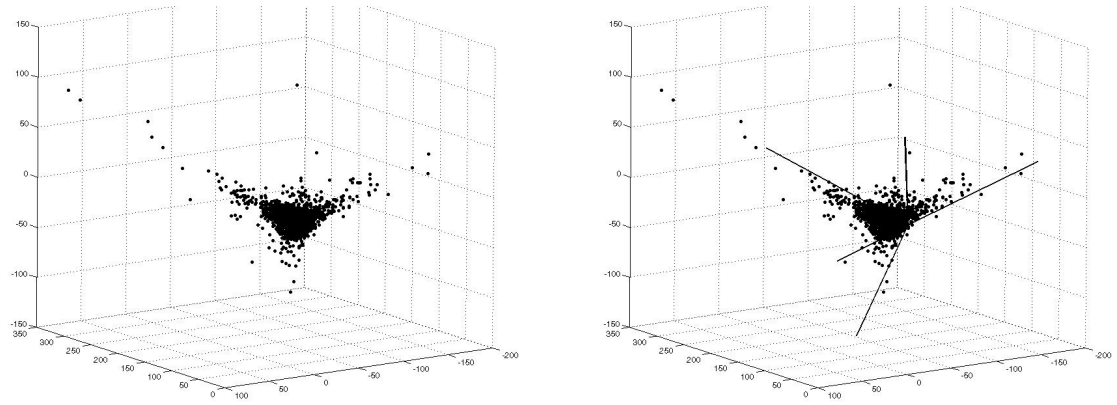

Figure 4.2: *left*: projection of the data onto its first three principal components; *right*: the first five basis vectors $\mathbf{H}_{1*}, \ldots, \mathbf{H}_{5*}$; Note that the principal components (extracted by PCA) do not coincide with the NMF basis components. They are used for illustration purposes only.

4.3, bottom). The same plot also reveals that the columns of $\mathbf{W}$ are roughly distributed according to the non-negative part of Gaussian distributions.
In figure (4.4), the extracted basis components $\mathbf{H}^{K=4}$ and $\mathbf{H}^{K=6}$ for $K = 4$ and $K = 6$ are contrasted with each other. Again, all components for the smaller decomposition (left) can be recovered in the decomposition into a larger number of components (right) ($\mathbf{H}^{K=4}_{1*} \approx \mathbf{H}^{K=6}_{1*}$, $\mathbf{H}^{K=4}_{2*} \approx \mathbf{H}^{K=6}_{2*}$, $\mathbf{H}^{K=4}_{3*} \approx \mathbf{H}^{K=6}_{3*}$ and $\mathbf{H}^{K=4}_{4*} \approx \mathbf{H}^{K=6}_{6*}$, roughly.) , plus two new basis vectors $\mathbf{H}^{K=6}_{5*}$ and $\mathbf{H}^{K=6}_{6*}$ appear. The same holds for the weight matrices, as shown in the scatterplots in the bottom of figure (4.4).
Summarizing, the basis vectors $\mathbf{H}$ and weight matrices $\mathbf{W}$ yield quite consistent results. The 3 dimensional projection given by figure (4.2) indicates that each basis vector $\mathbf{H}_{k*}$ points into one of the directions of the data agglomerations which point outwards from the origin in a shape resembling a star. The histograms of the columns of the weight matrices resemble non-negative parts of Gaussian distributions, sometimes exponential distributions.

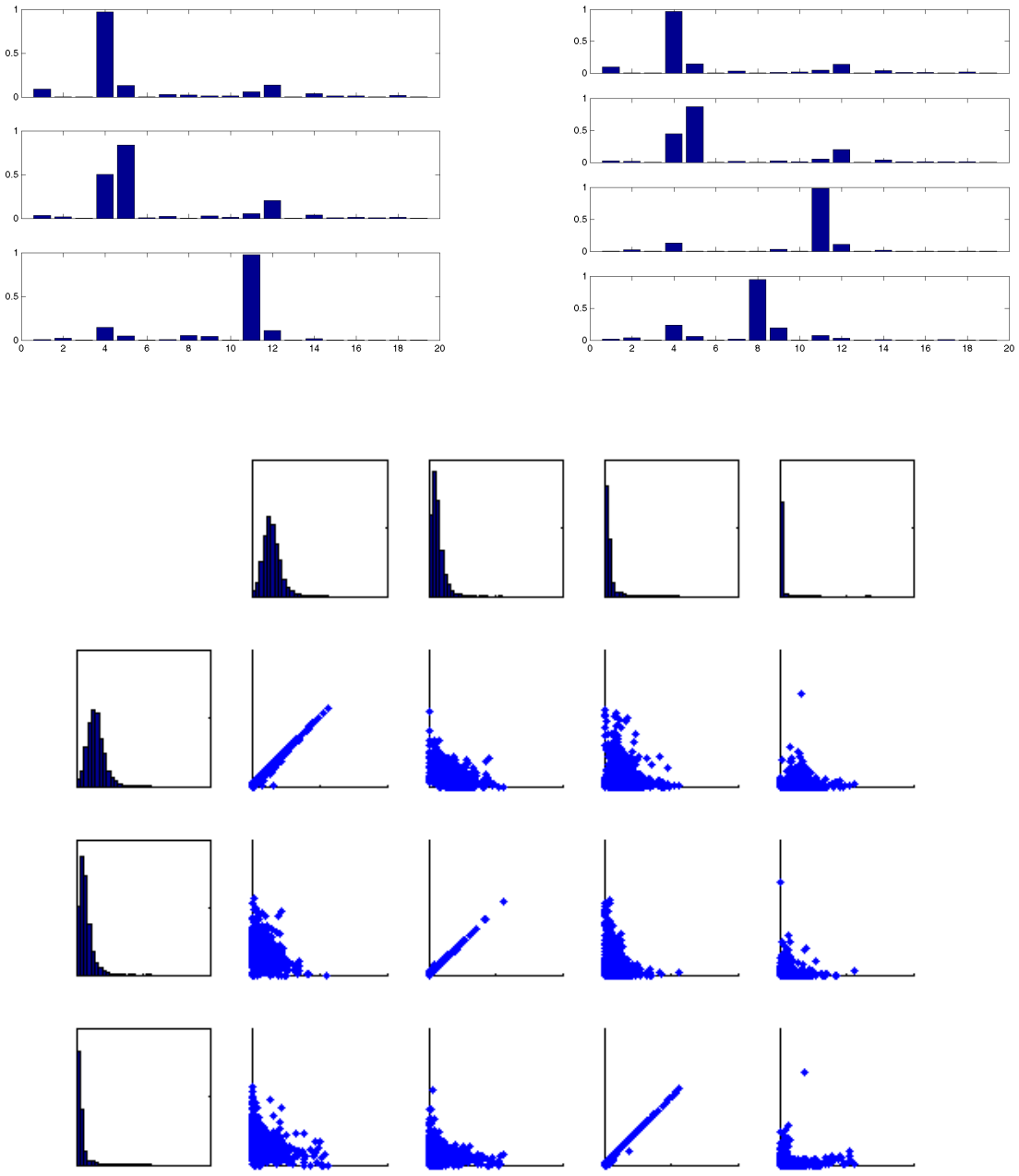

Figure 4.3: Different NMF decompositions using $K = 3$ and $K = 4$ components. *top*: The three basis vectors $\mathbf{H}^{K=3}$ (left) are quite similar to the first three rows of $\mathbf{H}^{K=4}$ (right). The fourth basis vector $\mathbf{H}^{K=4}_{4*}$ is relatively new, but has some correlations with $\mathbf{H}^{K=3}_{3*}$. *bottom*: Scatterplots from left to right: columns of $\mathbf{W}^{K=4}$; from top to bottom: columns of $\mathbf{W}^{K=3}$; Additionally, the histograms of each column are displayed.

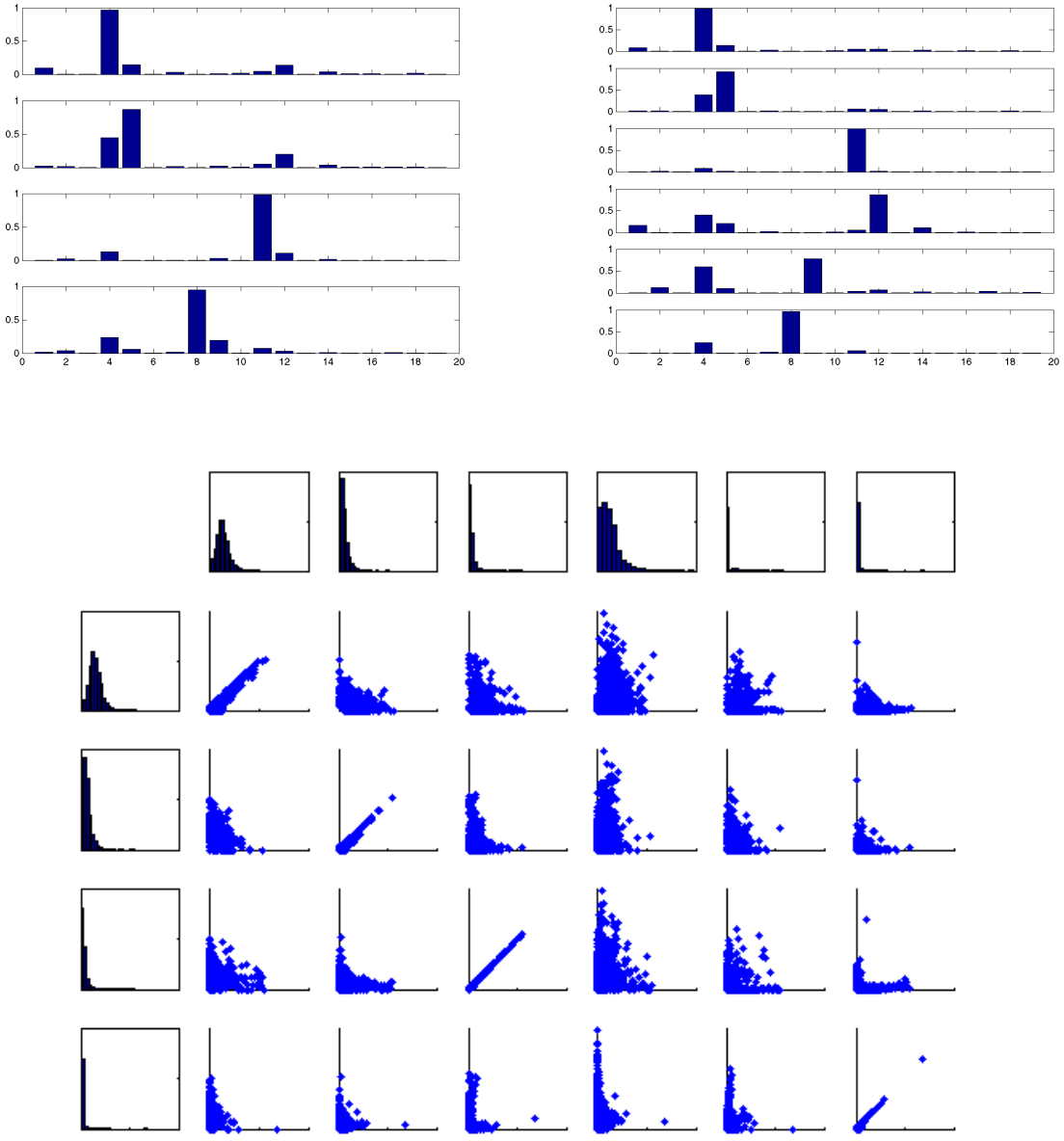

Figure 4.4: Comparison of the decompositions in $K = 4$ and $K = 6$ components; *top,left*: $\mathbf{H}^{K=4}$; *top,right*: $\mathbf{H}^{K=6}$; *bottom*: scatterplots of the columns $\mathbf{W}_{*k}^{K=4}$ vs. $\mathbf{W}_{*k}^{K=6}$

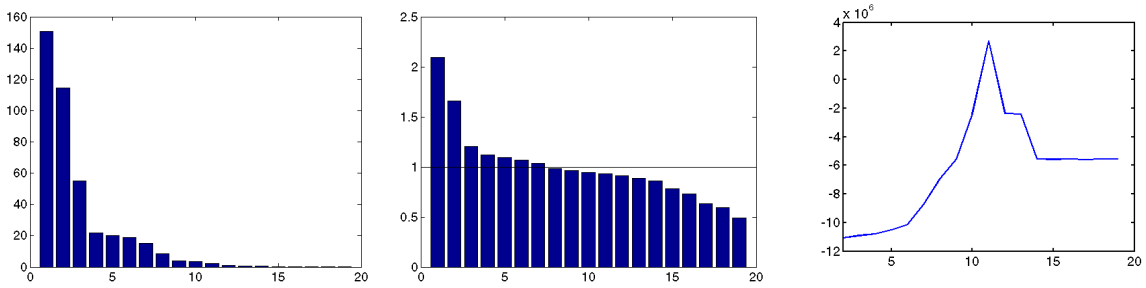

Figure 4.5: Eigenvalues of the covariance matrix (left) and correlation matrix (center) in descending order. The right plot shows the number of components K versus a variational Bayes criterion $\mathcal{B}_q$ which has a maximum at $K = 11$ (see text and paragraph 8.2.2 for details)

4.2.2 Number of components

The actual *best* number of components is a very hard question in the present example. As mentioned in the introductory NMF chapter 2, we assume the number of components K to be given, since it is required as input for usual NMF algorithms. Of course, in real world applications we do not know in advance how many underlying components there are in a data set.
Sometimes, the eigenvalue spectrum of the covariance or correlation coefficient matrix can give a hint on the dimensionality of the data space. In figure 4.5, a plot of the eigenvalues of the data covariance and correlation coefficient matrix is given.
Using normalized data such that the variance within a variable is 1, each observed variable contributes one unit of variance to the total variance. According to Kaiser's criterion [Kai58] each principal component which explains at least as much variance as one observed variable contains relevant information. In figure (4.5, center plot), there are seven eigenvalues greater than 1, so Kaiser's criterion suggests $K = 7$ to be the actual dimensionality.
In contrast, optical inspection of the eigenvalue spectrum of the data covariance matrix (4.5, left) may suggest either 3 principal components, 7,8, or 11 important components in the data (indicated by large decrease between adjacent eigenvalues).
Obviously, these rule of thumbs for the estimation of the dimensionality by judgment of the PCA eigenvalue spectrum does not lead to a clear statement on the actual number of components in the case considered here.
Anticipating the result of a later chapter, a variational Bayes procedure suggests that $K = 11$ components give the best explanation for this dataset (see the maximum of the variable $\mathcal{B}_q$ in figure 4.5, right) We will explain the theory behind the computation of this Bayesian criterion in detail in paragraph (8.2.2).

4.3 Summary

This section demonstrated the performance of the NMF technique on wafer test data which was aggregated in relative BIN counts per wafer. Each column of the data $\mathbf{X}$ corresponds to one BIN category, while each row constitutes the data of one wafer. Data entry X_{ij} is an approximate probability that a chip on wafer i fails in BIN category j. We assumed a linear non-negative superposition of K individual failure causes, ignoring all nonlinear effects which can be contained in such wafer test data.
The NMF methodology was proven to extract consistent components of a data set comprising $N = 2800$ wafers and $M = 19$ BIN categories. For different numbers of components K, rather the same

components are extracted, which was verified by visual inspection by the estimated basis components $\mathbf{H}_{k*}$ and scatterplots of the weights $\mathbf{W}_{*k}$.
The benefit of the NMF methodology to the overall failure analysis is to offer an alternative data representation, separated by potential causes and their individual contributions.
We do not discuss further analysis steps which are necessary to detect the actual root causes in the processing line. As sketched in section (1.2) other data analysis tools can be used to match the NMF findings with other historical data of the investigate wafers.
The determination of the actual number of hidden components remains an open issue. A sneak preview on a later chapter was given which addresses the problem of model order selection.
In this chapter, test data was analyzed on wafer level, i.e. fail probabilities in different BIN categories were approximated by the fraction of chips per wafer carrying the respective BIN labels, and a linear non-negative superposition model was assumed. In that case, usual NMF techniques can be applied. Due to the developments in chapter 3, the uniqueness ambiguity for a fixed number of components does not occur since we used a volume-constrained NMF algorithm.
While this chapter concerned the direct application of NMF to suitably aggregated data on wafer level, the next chapter will follow a different approach. In chapter 5, we will construct a non-negative superposition model for binary test data on chip level and develop a new extension of NMF suited to this problem.

Chapter 5

NMF extension for Binary Test Data

In this chapter a new approach to the analysis of wafermaps is introduced. In contrast to the preceding chapter (4), where test data on wafer level was approximated by a standard NMF model, a new method for binary data on chip level is developed here. The method is called *binNMF* and is aimed to decompose binary wafer test data into elementary root causes and their individual contributions. The data is assumed to be generated by a superposition of several simultaneously acting sources or elementary causes which are not observable directly. Throughout this chapter we will use the term *superposition* as a synonym for the action of several underlying sources and their joint effect on the outcome of the wafer test procedure.
Based on a minimum of assumptions the superposition process is modeled and its reversion allows to identify the underlying source characteristics.

5.1 Binary Data Sets

Binary test data arise for example in the final functionality tests of microchip fabrication. Irrespective of which physical quantity is actually measured, the information transmitted for every single die is a binary *pass* or *fail* variable. Hence already after the first failed single test the corresponding chip is labeled fail while it is labeled pass only if it passes all single tests. Such large binary data sets, which are collected in many practical applications apart from semiconductor industry, may be considered to be generated by diverse hidden underlying processes. Here, a multiple cause model is assumed for data generation, and the problem is related to NMF methodology. Not surprisingly, parallels exist between the proposed method and existing techniques from various disciplines such as statistics, machine learning, neural networks and bioinformatics which have strong overlap and use basically similar procedures for different target settings.

5.1.1 Defect patterns

Semiconductor microchips are usually processed on a silicon wafer. Before completion, each wafer passes up to several hundreds of processing steps. Most functionality tests can only be performed after the final processing step when the devices are completed. In general, the test outcome reflects whether a set of specifications is met (*pass, 0*) or not (*fail, 1*). Due to the diversity of processing steps like etching, implantation, photo lithography etc., here, the general term *'defect'* describes some malfunction dependent on one or more processing steps which render a manufactured device not to

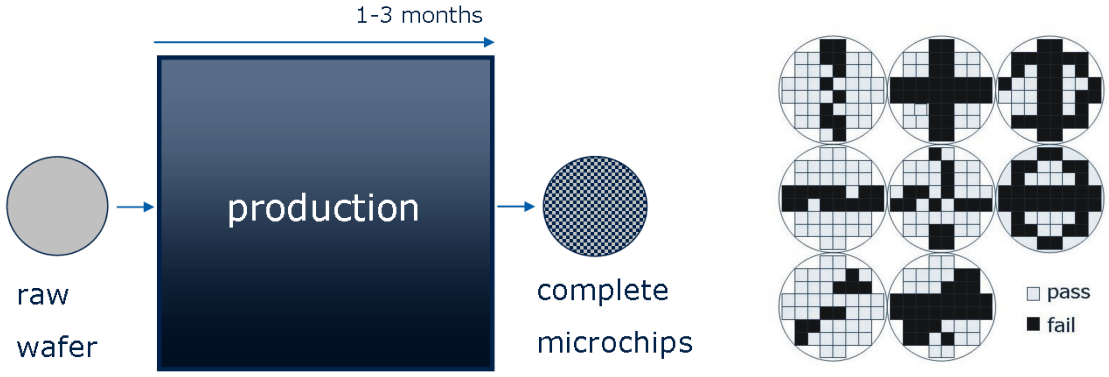

Figure 5.1: *left* : A raw wafer passes up to several hundreds of processing steps before the complete chips can be tested on their functionality *right*: In a set of binary wafermaps containing pass/fail information for the individual chips, sometimes localized failure patterns are present.

meet all defined performance and functionality specifications. Such a defect can be induced in any of the single manufacturing steps, or be the result of a combination of some not perfectly calibrated or matched instances, or even be already present in the raw material. Whatever is the number of physical quantities tested for every chip, finally each chip is characterized by a single binary variable $X_{ij} \in \{0, 1\}$ corresponding to *pass* or *fail* which depends on the margin of the actual test setup. This set of binaries for a whole wafer can be seen as a *wafer map*. Visual inspection of a large enough set of *wafer maps* often reveals geometrically localized structures, called failure patterns, on the disk such as rings, scratches or clusters.
The aim of the proposed exploratory *binNMF* data analysis methodology is the identification of failure patterns and to support the subsequent clarification of their hidden causes.

5.1.2 Data generating process

Imagine there are K possible elementary causes for a defect, and a combination of them is responsible for the actual test outcome. The superposition of these hidden causes reflects the following properties:

- Each single cause can separately increase the total fail probability.
- None of the single causes weakens the effect of the other causes.
- The overall fail probability is small only if all individual fail probabilities are small as well.

Now assume that M variables are measured simultaneously. The i-th observation then constitutes a row vector $\mathbf{X}_{i*} = (X_{i1}, \ldots, X_{iM})$ (see Fig. 5.1, right) whose components represent the binary *pass* ($X_{ij} = 0$) or *fail* ($X_{ij} = 1$) status of the M measurements. Here, each variable corresponds to a chip position (a specific x-y-coordinate one the wafer) and is indexed by j, and each observation i to a wafer.
Two additional assumptions are the following:

- To each source or hidden cause can be assigned a characteristic, locally resolved pattern of fail probabilities, henceforth called *basic pattern*. Some regions are assigned a larger, others are assigned a smaller fail probability reflecting the characteristics of the related hidden cause or source.
- Each basic pattern can vary its intensity only as a whole.

5.2 NMF for Binary Datasets

After having explained the meaning of the general term *defect* here in section 5.1.1, and stated the desired properties of the data generating process, we proceed to explain how the defect probabilities of M chips on N wafers can be described by means of two non-negative matrices $\mathbf{W}$ and $\mathbf{H}$.

5.2.1 The superposition approximation

Consider a finite number K of hidden sources or elementary failure causes $S_k,\ k = 1, \ldots, K$ underlying the observed failure patterns. The basic patterns related with these hidden sources are assumed to superimpose to the observed failure patterns constituting the wafermaps. The event *'chip j on wafer i is pass'* is encoded by $X_{ij} = 0$ and its conditional probability given hidden source S_k reads $P(X_{ij} = 0|S_k)$. The opposite event *'chip j on wafer i is fail'* is encoded as $X_{ij} = 1$ and its conditional probability given hidden source S_k reads $P(X_{ij} = 0|S_k)$, accordingly. The corresponding overall conditional *pass* and *fail* probabilities then factorize in case of independent single events and are given by

$$P(X_{ij} = 0|S_1, \ldots, S_K) = \prod_{k=1}^{K} P(X_{ij} = 0|S_k) \tag{5.1}$$

$$P(X_{ij} = 1|S_1, \ldots, S_K) = 1 - \prod_{k=1}^{K} P(X_{ij} = 0|S_k) \tag{5.2}$$

The factorial form of the overall conditional *pass*-probability given in eq. (5.1) reflects the superposition assumption: The overall conditional pass-probability can only be large, if none of its factors $P(X_{ij} = 0|S_k)$ is small. On the other hand, one single conditional *pass* probability close to zero given a specific failure cause S_k renders the overall conditional *pass* probability small. Thus, a chip is likely categorized as *fail* if at least one source S_k exists such that $P(X_{ij} = 0|S_k)$ is close to zero. But it is likely categorized as *pass* only if for all $S_k,\ k = 1, \ldots, K$ the conditional probabilities $P(X_{ij} = 0|S_k)$ are large.

5.2.2 The Poisson yield model

The Poisson yield model is a standard assumption in modeling semiconductor defect probabilities (see [Pha06]). Assuming that the defects are randomly distributed and that the occurrence of a defect at any location is independent of any other defect, the probability of having t defect events is given by the Poisson probability distribution

$$P_\lambda(t) = \exp(-\lambda)\frac{\lambda^t}{t!} \tag{5.3}$$

where the parameter λ reflects the expected number of defects per part and is calculated as the product of an area and a defect density.
Now assume a collection of N wafers, each containing M chips, arranged in an $N \times M$ data matrix $\mathbf{X}$. Assume further that there are K independent elementary causes, such that the probability of observing t defects related to cause k on chip j of wafer i can be expressed as

$$P_{\lambda_{ijk}}(t) = \exp\left(-\lambda_{ijk}\right)\frac{\lambda_{ijk}^t}{t!} \tag{5.4}$$

Given the superposition approximation discussed above, it is only needed to distinguish between the two cases of no defect, rendering the conditional *pass* probability high for all hidden sources, or at least one defect, rendering at least one conditional *pass* probability low on a wafer.
By now, the latent sources, chip positions and wafers are seen as independent. In the following, we introduce some dependency structure by some basic assumptions.

1. The k-th basic pattern $\mathbf{H}_{k*} = (H_{k1}, \ldots, H_{kM})$ is denoted by an M-dimensional row vector whose j-th component $H_{kj} \geq 0$ indicates the defect probability of chip j induced by source k.
2. Each elementary failure cause can contribute differently to the failure patterns of the different wafer maps. We describe the impact of source k on wafer i by the weight factor $W_{ik} \geq 0$, where $W_{rk} < W_{sk}$ means that source k has less impact on wafer r than on wafer s.
3. We finally assume that a whole set of N wafer maps, each containing M chips, can be represented by a finite number of K elementary failure causes. Each observed wafer map represents a realization of an individual superposition process of the same underlying elementary failure causes.

Given independent elementary failure patterns $\mathbf{H}_{k*}$ and the superposition approximation, the conditional *pass*-probability of chip j on wafer i can be expressed as

$$P(X_{ij} = 0|S_k) = \exp\left(-W_{ik}H_{kj}\right) \tag{5.5}$$

where, according to eq. (5.4), $\lambda_{ijk} = W_{ik}H_{kj}$ and $t = 0$. The conditional *pass* and *fail* probabilities according to the factorial model in eq. (5.1) now read

$$P(X_{ij} = 0|S_1, \ldots, S_K) = \prod_{k=1}^{K} \exp\left(-W_{ik}H_{kj}\right) = \exp\left(-\sum_{k=1}^{K} W_{ik}H_{kj}\right) \tag{5.6}$$

$$P(X_{ij} = 1|S_1, \ldots, S_K) = 1 - \exp\left(-\sum_{k=1}^{K} W_{ik}H_{kj}\right) \tag{5.7}$$

where we recognize the exponent in the last expression as (i, j)-th entry of the matrix product $\mathbf{WH}$ which corroborates the close relationship to NMF techniques. Since all W_{ik}, H_{kj} are non-negative by assumption, the conditional *fail* probability becomes a linear function of the underlying causes, i.e. $P(X_{ij} = 1|S_1, \ldots, S_K) \approx \sum_k W_{ik}H_{kj}$ for small arguments, and saturates at $P(X_{ij} = 1|S_1, \ldots, S_K) \approx 1$ for large arguments.

5.2.3 Bernoulli Likelihood

Denoting the Bernoulli-parameter $\Theta_{ij} = P(X_{ij} = 1|S_1, \ldots, S_K) = 1 - \exp\left(-[\mathbf{WH}]_{ij}\right)$, the Bernoulli likelihood of entry (i, j) is

$$P(X_{ij}|\Theta_{ij}) = \Theta_{ij}^{X_{ij}}(1 - \Theta_{ij})^{1-X_{ij}} \tag{5.8}$$

Assuming that the data are independent, given $\mathbf{\Theta}$, this leads to an overall log-likelihood

$$LL = \sum_{i=1}^{N}\sum_{j=1}^{M} \left\{X_{ij} \ln(1 - \exp\left(-[\mathbf{WH}]_{ij}\right)) - [\mathbf{WH}]_{ij} + X_{ij}[\mathbf{WH}]_{ij}\right\} \tag{5.9}$$

which has to be maximized in the variables $\mathbf{W}$ and $\mathbf{H}$ with respect to the non-negativity constraints $\mathbf{W}, \mathbf{H} \geq 0$.

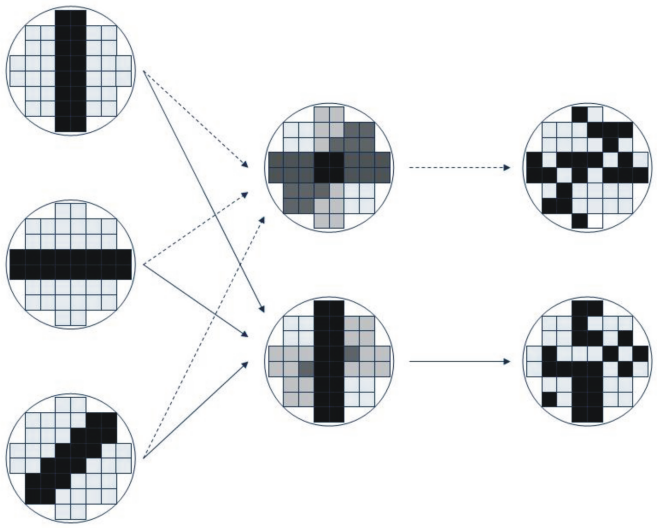

Figure 5.2: Illustration of the superposition model. *left:* Elementary failure patterns $\mathbf{H}_{1*}, \mathbf{H}_{2*}, \mathbf{H}_{3*}$ describing the characteristics of three hidden sources S_1, S_2, S_3 *center:* 2 different superpositions of fail probabilities $P(\mathbf{X}_{1*} = 1|S_1, S_2, S_3)$ and $P(\mathbf{X}_{2*} = 1|S_1, \ldots, S_3)$ *right:* observable binary realizations $\mathbf{X}_{1*}$, $\mathbf{X}_{2*}$. Note that in general the elementary failure patterns $\mathbf{H}$ are continuous-valued and not restricted to binary values

binNMF summary

N :	number of objects	$\in$	$\mathbb{N}$
M :	dimension	$\in$	$\mathbb{N}$
K :	number of basic patterns	$\ll$	$\min(N, M)$
$\mathbf{X}$	data matrix		$N \times M$
$\mathbf{W}$	coefficient matrix		$N \times K$
$\mathbf{H}$	pattern matrix		$K \times M$
$\mathbf{X}_{ij}$	$\begin{cases} = 0: \text{ 'pass'} \\ = 1: \text{ 'fail'} \end{cases}$	$\in$	$\{0, 1\}$
$\mathbf{W}_{ik}$	weight of pattern k in object i	$\in$	$[0, W_{max}]$
$\mathbf{H}_{kj}$	intensity on position j in pattern k	$\in$	$[0, 1]$
$P(\mathbf{X}_{ij} = 0 \mid S_1, \ldots, S_K)$	$e^{-[\mathbf{WH}]_{ij}}$	$\in$	$]0, 1]$
$P(\mathbf{X}_{ij} = 1 \mid S_1, \ldots, S_K)$	$1 - e^{-[\mathbf{WH}]_{ij}}$	$\in$	$[0, 1[$

Figure 5.3: Summary of the constituents of the*binNMF* model: binary data $\mathbf{X}$ is modeled by a nonlinear function of a product of non-negative matrices $\mathbf{W}$ and $\mathbf{H}$ expressing the conditional probabilities according to K underlying hidden sources $S_1, \ldots, S_K$

Note that the Bernoulli likelihood is a natural choice for binary data (see e.g. [Tip99], [CDS01], [BKF09]).
In the appendix (B), experiments with an alternative family of cost functions based on the Minkowski-R measure [Bis96] are reported.

5.3 Optimization strategies

Although if the task of maximizing the log likelihood (5.9) can be interpreted as the minimization of a special NMF cost function, the Bernoulli likelihood 5.9 has not been a topic in the NMF literature so far and we can not apply any of the NMF algorithms discussed in chapter 2 directly for the optimization. However, the challenges of an optimization under non-negativity constraints are similar.
Several alternating optimization schemes were tested which, at each step, keep one parameter matrix fixed and update the other one according to some gradient rule with respect to the non-negativity constraints on $\mathbf{W}$ and $\mathbf{H}$. In principle, any of the NMF optimization strategies discussed in chapter 2 can be tested on the Bernoulli likelihood although the resulting algorithms may be more complex than for usual NMF cost functions. In general, Newton type algorithms have a greater computational complexity in each iteration, while leading to a comparative large increase of the log likelihood per iteration. Since especially the logarithm in (5.9) induces many local optima, the steps taken by Newton type algorithms are very often such as to get stuck in a local optimum in the early iterations. For this reasons, we concentrate on simpler optimization strategies which make smaller but simpler steps in each iteration. We discuss an alternating gradient ascent and a multiplicative update scheme.

5.3.1 Alternating Gradient Ascent

The basic gradient ascent scheme is given by

$$W_{ik} \leftarrow W_{ik} + \eta_W \frac{\partial LL}{\partial W_{ik}} \tag{5.10}$$

$$H_{kj} \leftarrow H_{kj} + \eta_H \frac{\partial LL}{\partial H_{kj}} \tag{5.11}$$

where the left hand sides must not become negative. For tiny step sizes η_W, η_H an actual increase of the log likelihood is guaranteed, but many iterations are necessary. On the other hand, a step too large can either lead to a decrease of LL or to negative elements in the parameter matrices.
The alternating gradient ascent algorithm consists of the following steps:

- `update` $\mathbf{W}$`:`
  1. `update` $\mathbf{W}$ `according to eq. (5.10)`
  2. `set negative elements in` $\mathbf{W}$ `to zero`
  3. `if` $LL^{new} < LL^{old}$ `diminish` η_W `and try again`
     `else slightly increase` η_W `and go on`
- `update` $\mathbf{H}$`:`
  1. `update` $\mathbf{H}$ `according to eq. (5.11)`
  2. `set negative elements in` $\mathbf{H}$ `to zero`
  3. `if` $LL^{new} < LL^{old}$ `diminish` η_H `and try again`
     `else slightly increase` η_H `and go on`

However, a general limitation of gradient optimization techniques is their getting stuck in local optima. In addition, once one of the stepsize parameters η_W, η_H has become very small, it takes very long time before reasonable progress can be made even if bigger steps were possible. Introducing separate stepsize parameters for each row $\mathbf{W}_{i*}$ and column $\mathbf{H}_{*j}$ can lead to better performance, but also requires more administration effort.

5.3.2 Multiplicative updates

The following simple multiplicative update rules do not require stepsize parameters and the evaluation of the log likelihood during the iterations:

$$W_{ik} \leftarrow W_{ik} \frac{\sum_j \frac{X_{ij} H_{kj}}{1-\exp(-[\mathbf{WH}]_{ij})}}{\sum_j H_{kj}} \tag{5.12}$$

$$H_{kj} \leftarrow H_{kj} \frac{\sum_i \frac{X_{ij} W_{ik}}{1-\exp(-[\mathbf{WH}]_{ij})}}{\sum_i W_{ik}} \tag{5.13}$$

Eqns. (5.12) and (5.13) are fix point-equations which strongly resemble the original Lee and Seung updates [LS01] for NMF as discussed in chapter 2. They share the benefits of not having to control a stepsize parameter and the non-negativity constraints. On the other hand, they also share the well known drawbacks of the Lee-Seung method: convergence towards a saddle-point can not be excluded, and, once a parameter is exactly zero, it remains zero.

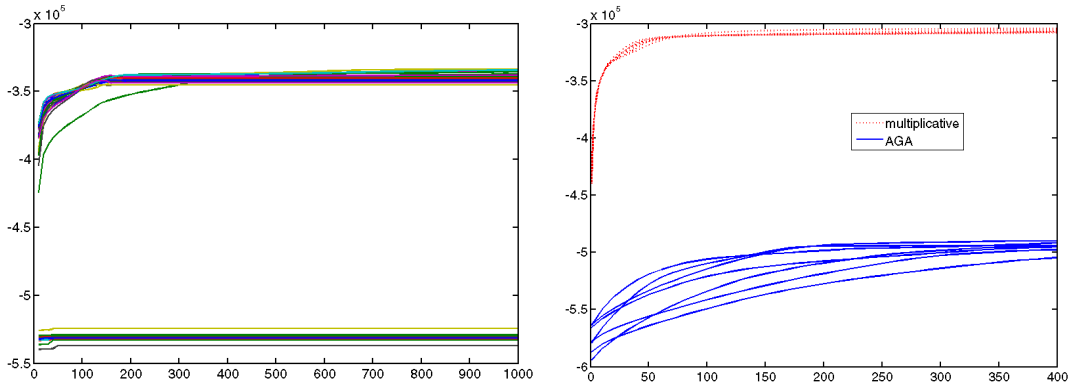

Figure 5.4: The Bernoulli log likelihood as a function of the number of iterations. *left*: performance of the alternating gradient algorithm (AGA) (sec. 5.3.1) in 50 randomly initialized runs. Depending on the initialization, several local maxima are reached, some of which in early iterations. *right*: Comparison between the multiplicative (sec. 5.3.2) and alternating gradient algorithm (sec. 5.3.1). Both algorithms were initialized by intermediate results of a preprocessing step (see sec. 5.3.4)

It is easy to verify that a local maximum of the Bernoulli log likelihood (5.9) is a fix point of the multiplicative updates given by eqns. (5.12, 5.13):
Setting the partial derivative w.r.t. H_{kj} to zero yields

$$
\begin{aligned}
0 &= \frac{\partial LL}{\partial H_{kj}} && (5.14)\\
&= \sum_i \left(\frac{X_{ij} \exp\left(-[\mathbf{WH}]_{ij}\right)}{1 - \exp\left(-[\mathbf{WH}]_{ij}\right)} - (1 - X_{ij}) \right) W_{ik} && (5.15)\\
&= \sum_i \left(\frac{X_{ij}}{1 - \exp\left(-[\mathbf{WH}]_{ij}\right)} - 1 \right) W_{ik} && (5.16)\\
\Leftrightarrow \quad & \sum_i W_{ik} = \sum_i \frac{X_{ij}}{1 - \exp\left(-[\mathbf{WH}]_{ij}\right)} && (5.17)\\
\Leftrightarrow \quad & \frac{\sum_i \frac{X_{ij}}{1-\exp(-[\mathbf{WH}]_{ij})}}{\sum_i W_{ik}} = 1 && (5.18)
\end{aligned}
$$

the case W_{ik} is analogous.
Although we can show that an optimum is a fixed point of the update equations, this is not an actual proof that the algorithm can even get close to such an optimum.
In practice, however, good convergence properties are observed.
Rewriting e.g. eq. (5.13) in an additive fashion leads to

$$
H_{kj} \leftarrow H_{kj} + \frac{H_{kj}}{\sum_i W_{ik}} \underbrace{\left[\sum_i \frac{X_{ij} W_{ik}}{1 - \exp\left(-[\mathbf{WH}]_{ij}\right)} - \sum_i W_{ik} \right]}_{= \frac{\partial LL}{\partial H_{kj}}} \qquad (5.19)
$$

where the underbraced term is the partial derivative $\frac{\partial LL}{\partial H_{kj}}$.

Convergence problems can only arise if the stepsize $\frac{H_{kj}}{\sum_i W_{ik}}$ were large. In that case, we could introduce a parameter ω and modify the update equations (5.12, 5.13) to

$$W_{ik} \leftarrow W_{ik} \left[\frac{\sum_j \frac{X_{ij} H_{kj}}{1-\exp\left(-[\mathbf{WH}]_{ij}\right)}}{\sum_j H_{kj}} \right]^{\omega} \tag{5.20}$$

$$H_{kj} \leftarrow H_{kj} \left[\frac{\sum_i \frac{X_{ij} W_{ik}}{1-\exp\left(-[\mathbf{WH}]_{ij}\right)}}{\sum_i W_{ik}} \right]^{\omega} \tag{5.21}$$

One can always choose ω small enough such that the updating factor is close to 1, and the change between successive values of H_{kj} is small. A general discussion on such multiplicative heuristic formulas can be found in [CZPA09], where ω represents a relaxation parameter, typically in the range $[0.5, 2]$.
As mentioned, in the experiments discussed here we set $\omega = 1$ and always observed good monotonicity and much faster convergence times compared to the alternating gradient algorithm (AGA) (see Fig. 5.4, right).

5.3.3 The noisy case

In real world applications, there is always some data which cannot be explained by any of the K assumed underlying sources. Such random noise can cause serious convergence problems or spoil some of the reconstructed basic patterns. We circumvent these problems by adding a dummy pattern $\mathbf{H}_{K+1,*}$ which consists of all ones. During optimization, we treat the corresponding weights $\mathbf{W}_{*,K+1}$ just as usual ones while keeping the pattern constant all the time. This simple trick offers the algorithm an opportunity to explain data points $X_{ij} = 1$ which do not suit the model of K sources well by increasing their noise parameter $W_{i,K+1}$. Without this precautionary measure, a single event $X_{ij} = 1$ which does not suit the model tends to increase the row $\mathbf{W}_{i*}$ or the column $\mathbf{H}_{*j}$ or both because of the logarithmic penalty $X_{ij} \ln(1 - \exp -[\mathbf{WH}]_{ij})$.

5.3.4 Preprocessing

Particularly the logarithm in eq. (5.9) can cause serious global convergence problems by inducing local maxima to the log-likelihood function. Any point $X_{ij} = 1$ with a small probability $1 - \exp\left(-[WH]_{ij}\right)$ will result in a logarithmic divergence of the log-likelihood. The optimization algorithm thus will try to compensate the divergence by increasing $[\mathbf{WH}]_{ij}$. In order to attenuate this problem, we propose an appropriate preprocessing step for our optimization procedure.
Introducing an auxiliary variable $\alpha \in]0, 1[$, we set

$$\begin{array}{lll} P(X_{ij} = 1 | S_1 \ldots, S_K) = 0, & \text{if} & X_{ij} = 0 \\ P(X_{ij} = 1 | S_1 \ldots, S_K) = \alpha, & \text{if} & X_{ij} = 1 \end{array} \quad \text{for all } i, j. \tag{5.22}$$

This can be summarized by

$$\alpha X_{ij} = 1 - \exp\left(-[\mathbf{WH}]_{ij}\right) \Leftrightarrow -\ln(1 - \alpha X_{ij}) = [\mathbf{WH}]_{ij} \tag{5.23}$$

Since the left hand side of the last equation is always nonnegative we recover a standard NMF problem $\mathbf{X}' \approx \mathbf{WH}$ when substituting $X'_{ij} =: -\ln(1 - \alpha X_{ij})$.
Choosing the squared Euclidean distance as a cost function

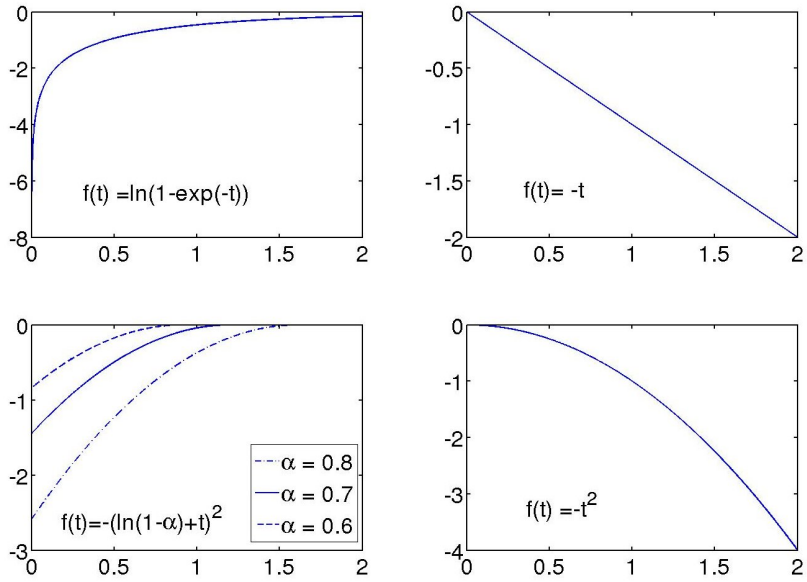

Figure 5.5: The preprocessing procedure approximates the penalties considered by the actual log likelihood (top row) for $X_{ij} = 1$ (left) and $X_{ij} = 0$ (right) by a quadratic form with an adjustable parameter α (bottom row).

$$E(\alpha, \mathbf{W}, \mathbf{H}) = \sum_{i=1}^{N} \sum_{j=1}^{M} (\ln(1 - \alpha X_{ij}) + [\mathbf{WH}]_{ij})^2 \tag{5.24}$$

the well-known Alternating Least Squares Algorithm as described in [CZA08] can be used to minimize (5.24) with respect to $\mathbf{W} \geq 0$ and $\mathbf{H} \geq 0$. The ALS-updates are given by

$$H_{rs} \leftarrow \max\{\epsilon, -\sum_{i=1}^{N} [(\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T]_{ri} \ln(1 - \alpha X_{is})\} \tag{5.25}$$

$$W_{lm} \leftarrow \max\{\epsilon, -\sum_{j=1}^{M} \ln(1 - \alpha X_{lj})[\mathbf{H}^T(\mathbf{HH}^T)^{-1}]_{jm}\} \tag{5.26}$$

In fact, any NMF algorithm could be used in the preprocessing step. Our choice of the ALS procedure is motivated by its simplicity and speed. The bad theoretical convergence properties (induced by truncation of negative elements) are alleviated by repeating the procedure several times using different random initializations for $\mathbf{H}$ and $\mathbf{W}$ and retaining only the solution with the smallest Euclidean distance. Multiple random initializations also lead to a more complete coverage of the search space.

Determining the Parameter α

The effect of α can be understood in the following way: As long as the matrix factorization framework permits it, an optimization algorithm should increase those $[\mathbf{WH}]_{ij}$ for which $X_{ij} = 1$, and diminish those $[\mathbf{WH}]_{ij}$ for which $X_{ij} = 0$ in order to reach the maximum likelihood solution. This kind of learning dynamics is also conserved in our simplified version, where optimal solutions tend towards

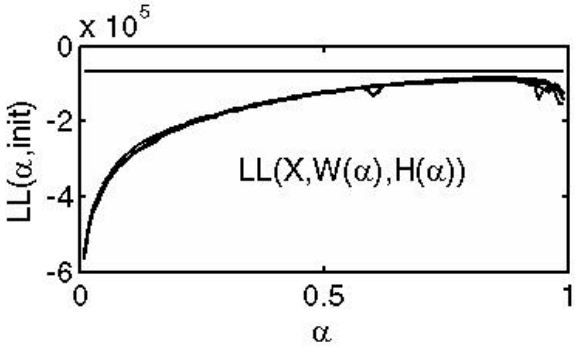

Figure 5.6: Log-likelihood of the approximations computed by the ALS-method as a function of α for 10 random initializations. The best value is obtained for $\alpha = 0.87$ in this example. The horizontal line denotes the true log-likelihood.

$-\ln(1-\alpha)$ for $X_{ij} = 1$ and 0 if $X_{ij} = 0$. Thus, qualitatively, the basic properties of the data are roughly reflected in the simplification. However, an optimal α cannot be estimated from the data directly but has to be determined by an additional optimization process.
The true quantity to be maximized is the log-likelihood function (5.9) which is a sum over NM individual terms. The costs corresponding to terms $X_{ij} = 1$ and $X_{ij} = 0$ are asymmetric for the two cases. A $X_{ij} = 1$ term leads to costs $\ln(1 - \exp(-[\mathbf{WH}]_{ij}))$, whereas a $X_{ij} = 0$ term yields a cost of $-[\mathbf{WH}]_{ij}$ (see Fig. 5.5, top row). The (negative of the) Euclidean cost function (5.24) implies a quadratic cost for both cases instead (see Fig. 5.5, bottom row). $X_{ij} = 0$ leads to a term $[\mathbf{WH}]_{ij}^2$, while $X_{ij} = 1$ yields a cost of $-\left(\ln(1-\alpha) + [\mathbf{WH}]_{ij}\right)^2$. The latter is an inverted parabola with a maximum at $-\ln(1-\alpha)(> 0)$. Note that only the left branch of this parabola needs to be considered, since the terms $X_{ij} = 1$ seek to approach the maximum at $[\mathbf{WH}]_{ij} = -\ln(1-\alpha)$ and the terms $X_{ij} = 0$ favour small values $[\mathbf{WH}]_{ij}$.
From simulations on toydata sets, we observed that the best obtained log-likelihood $LL(\mathbf{X}, \mathbf{W}(\alpha), \mathbf{H}(\alpha))$ among several randomly initialized runs resembles a concave function of α (see Figure 5.6). Thus, a Golden Section Search procedure can be applied to obtain the optimal α in a reasonable amount of trials and computational time.
In summary, with the help of the auxiliary parameter α, the original log-likelihood components for $X_{ij} = 1$ and $X_{ij} = 0$ are approximated by quadratic forms. This parameter α has to be optimized in order to find the best such approximation for a given dataset.

5.3.5 Uniqueness

Referring to the non-uniqueness of unconstrained NMF solutions, in case of binary datasets this problem is relieved.
As is shown in a schematic drawing in Fig. 5.7, with continuous-valued data several equivalent solutions exist because the spanning basis vectors $\mathbf{H}_{1*}, \ldots, \mathbf{H}_{K*}$ can lie anywhere between the data cloud and the boundaries of the non-negative orthant (see also chapter 3 and [SPTL09]). In a binary problem setting, the data lie in the corners of a M-dim hypercube in the non-negative orthant having one corner at the origin and edge length $-\ln(1-\alpha)$. The (continuous-valued) basis vectors $\mathbf{H}_{k*}$ coincide with the borders of this hypercube if $K = M$ and are inside the hypercube if $K < M$. Thus, the additional freedom of the basis vectors due to multiple possibilities outside the data cloud in the continuous case is missing in the binary situation, since the basis vectors are inside the data cloud in this case.

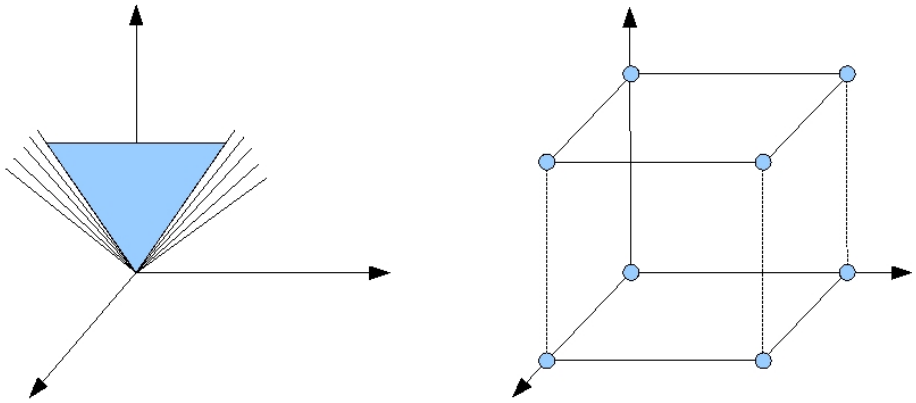

Figure 5.7: *left*: Non-uniqueness of NMF solutions for continuous-valued data illustrated with a 2-dim manifold embedded in 3-dim space. Different solutions are indicated by bundles of spanning basis vectors $\mathbf{H}_{1*}, \mathbf{H}_{2*}$ *right*: In case of binary data, there is no such ambiguity since the data lie in the corners of a hypercube with edge length $-\ln(1-\alpha)$. Note that one of the corners corresponds to the origin and the edges span the positive orthant.

We can summarize the whole optimization strategy as searching the parameter space by repetitive running a fast ALS algorithm on a simplified problem involving an additional parameter α. This α is optimized by a Golden Section Search procedure. The simplified problem is solvable by standard NMF algorithms. Considering the results of chapter 3, NMF is known not to produce unique results without additional restrictions in general. In the special case of binary data, however, these uniqueness problems do not exist. Thus the preprocessing procedure for fixed α leads to optimal solutions of $E(\alpha, \mathbf{W}, \mathbf{H})$ in eq. (5.24). Optimization of α then leads to a good approximation of the original problem $LL(\mathbf{W}, \mathbf{H})$ in eq. (5.9). From this point, we can run the AGA or multiplicative algorithm to optimize the actual log likelihood LL into the nearest local maximum. In that sense we can interpret the binary NMF problem to have a quasi optimal solution inherited from the quadratic approximation.

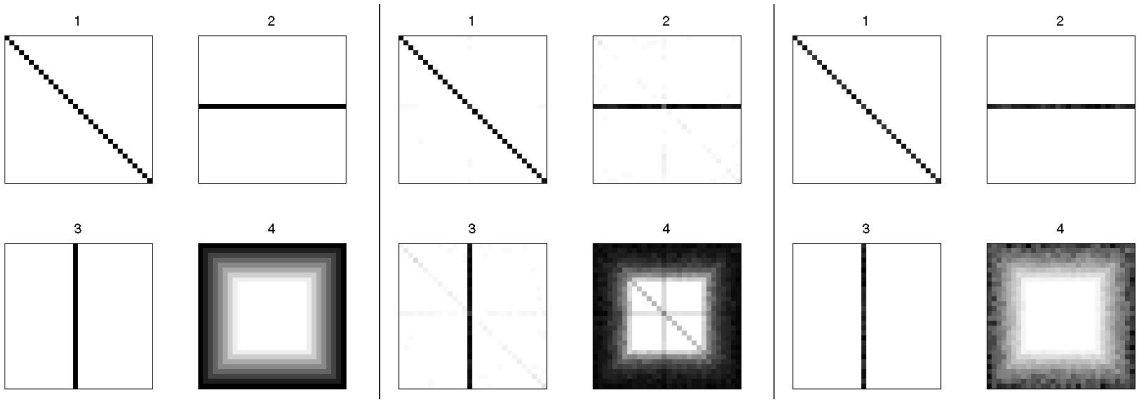

Figure 5.8: *left*: 30×30 source patterns $\mathbf{H}_{1*}^{orig} \ldots, \mathbf{H}_{4*}^{orig}$ valued in $[0, 1]$ (white:0, black:1, grey: intermediate). *center*: Reconstructions gained via the ALS preprocessing $\mathbf{H}^{ALS}$, *right*: Maximum likelihood solutions $\mathbf{H}^{ML}$ gained vie gradient refinement

5.4 Simulations

A 1000×900 binary data matrix $\mathbf{X}$ was created by setting the (i,j)-th entry to 1 with probability

$$p_{ij} = 1 - \exp\left(-[\mathbf{W}^{orig}\mathbf{H}^{orig}]_{ij}\right) \tag{5.27}$$

(see Fig. 5.9 for examples) using $K = 4$ fixed failure patterns $\mathbf{H}_{1*}^{orig}, \ldots, \mathbf{H}_{4*}^{orig}$ and a randomly generated 1000×4 coefficient matrix $\mathbf{W}^{orig}$.
We use three binary patterns (white: 0, black: 1) and one pattern of values graded from zero in the center to one on the edges (see Fig. 5.8, left hand side).
The preprocessed ALS-method yields quite good approximations of the original source patterns in this example. After 1000 iterations refinement by Alternating Gradient Ascent, nearly perfect reconstruction of the original patterns is achieved (see Fig. 5.8). Note that in the images $\mathbf{W}$ and $\mathbf{H}$ are re-scaled such that the maximum value in each pattern $\mathbf{H}_{k*}$ is given by one for reasons of better visualization.
The top row of Fig. 5.9 contains examples for coefficients W_{ik} in the toy data set. The second row shows the corresponding binary images $\mathbf{X}_{i*}$ which are obtained by a setting the entries to 1 with probabilities given by eq. (5.27). The leftmost example $\mathbf{X}_{1*}$ consists of the vertical line ($\mathbf{H}_{3*}^{orig}$) and the continuous pattern on the edge ($\mathbf{H}_{4*}^{orig}$), weighted by W_{13} and W_{14}, while $W_{11}, W_{12} = 0$. The rightmost example $\mathbf{X}_{8*}$ consists of the diagonal, horizontal and vertical lines ($\mathbf{H}_{1*}^{orig}$,$\mathbf{H}_{2*}^{orig}$ and $\mathbf{H}_{3*}^{orig}$), weighted by W_{81}, W_{82} and W_{83} while $W_{84} = 0$.
The *binNMF* algorithm receives only the binary realizations $\mathbf{X}$ and the information $K = 4$ as input and separates the underlying patterns $\mathbf{H}$ and the weights $\mathbf{W}$.
The third row of Fig. 5.9 shows the weights $\mathbf{W}^{ALS}$ (the solution with optimal $E(\alpha, \mathbf{W}, \mathbf{H})$ which are the input for the refinement by Alternating Gradient Ascent (solutions $\mathbf{W}^{ML}$ shown in the last row).

5.5 Real world application

The examples presented show decompositions of real world data sets which were derived from special measurements. The potential of the *binNMF* technique to discover localized patterns on binary wafermaps is demonstrated.

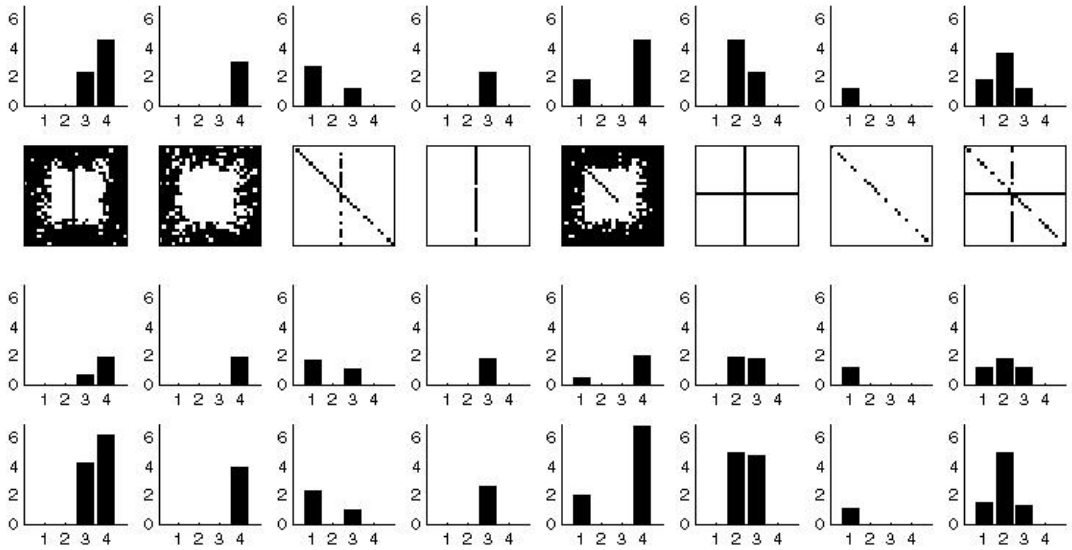

Figure 5.9: Toydata examples. Top: Original coefficients $\mathbf{W}_{i*}$, Second row: Binary realizations $\mathbf{X}_{i*}$, Third row: Coefficients gained by ALS, Bottom: Coefficients after refinement by Gradient Ascent

5.5.1 Real World Example I

The real world example shows a decomposition of $M = 3043$ wafers, each containing $M = 500$ chips into $K = 6$ basic patterns (see Fig. 5.10). The data stems from measurements which aim to identify latent structures and detect potential failure causes in an early stage of the processing chain. A different decomposition of the same dataset is shown in [SPL10]. The estimated $K = 6$ basic patterns $\mathbf{H}_{1*}, \ldots, \mathbf{H}_{6*}$ have clearly different characteristics: One pattern of higher *fail* probability on the upper side of the wafer, a bead centered on the wafer, a ring of fails on the edge zone, two different repeated structures, and defects concentrated on the bottom of the wafer were detected. The related weight coefficients $\mathbf{W}_{*1}, \ldots, \mathbf{W}_{*6}$ store the activity of each of the 6 putative basic patterns on each wafer separately. This new representation of the data highlights wafers affected by the detected sources and is intended to support the detection of potential error causes.
Figure (5.11) illustrates examples of the parts-based representation by relevant underlying patterns of *fail* probabilities $1 - \exp(-W_{ik}H_{k*})$ which generate the observed failure patterns. In this respect, the *binNMF* method provides an alternative representation of a given dataset, separated in terms of putative sources and their individual contributions.

5.5.2 Real World Example II

This second example shows two different decompositions of a total of 760 wafers into $K = 5$ and $K = 6$ source components (see Fig. 5.12). Each wafer contains 3500 chips here and hence the images have a higher resolution than in the previous example.
Again, different kinds of patterns are detected, some of which seem to have some rotational symmetry. An interesting feature is visible in these images: In wafer processing, a set of 50 or 25 wafers is treated as a unit called *lot*. Usually, all wafers of a lot have the same processing history, e.g. the whole lot is processed by one machine at once, while different lots can have a different process history. In this example, there are 25 wafers in a lot and we see that the computed source patterns often are active in bundles of about 25 wafers. Either a whole lot is affected by a source or not. Note that the upper

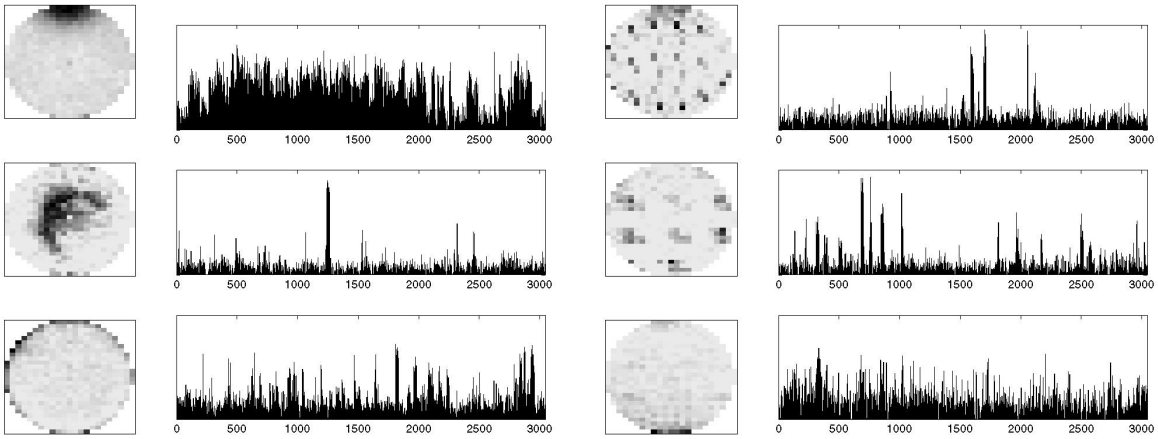

Figure 5.10: Estimated basic patterns $\mathbf{H}_{1*}, \ldots, \mathbf{H}_{6*}$ and contribution coefficients $\mathbf{W}_{*1}, \ldots, \mathbf{W}_{*6}$, estimated from a real dateset comprising 3043 wafers and 500 chips.

leftmost pattern 1 and the middle leftmost pattern 2 in decomposition in $K = 6$ sources shown in the bottom of figure (5.12) seem to be rotated versions of each other and that source 1 becomes active just as source 2 becomes inactive (corresponding to values between 350 and 375 on the abscissa which give the indices of the processed wafers i.)
In this example two different decompositions in $K = 5$ and $K = 6$ sources are shown to emphasize that there is no built-in criterion to automatically judge an optimal number of sources.
Based on the decompositions gained by the *binNMF* technique, engineers can select either already known patterns or previously unknown ones and use the weight matrices for further analysis.
The main advantage of the technique is that the actual production process is treated as a black box. As mentioned in the introductory chapter 1.2, there are lots of additional data available and can be matched with the information generated by the *binNMF* technique to allow previously impossible insights and improve failure analysis.
Note that the geometrical structure of the extracted patterns is a feature which can be easily assessed by human experts. This can be used as an independent control mechanism to judge individual decompositions, since the *binNMF* technique does not exploit the geometrical structure of the patterns. The algorithm would reproduce the results irrespective of the ordering in which the chip positions are aligned in the data matrix.

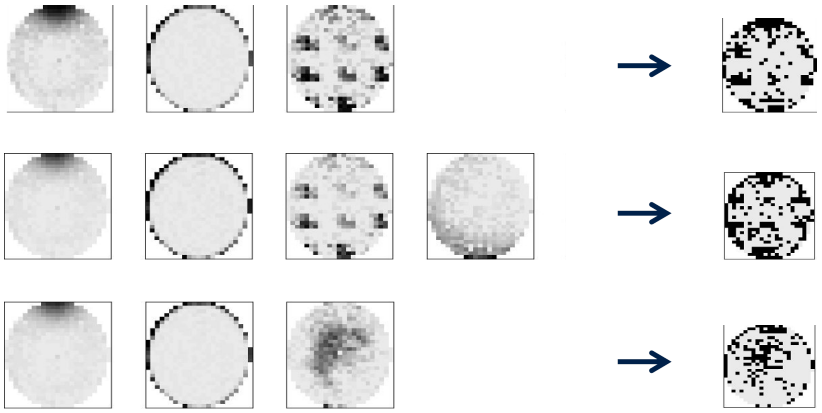

Figure 5.11: Examples for parts based representations obtained by *binNMF*. The rightmost wafermaps are the binary original data, the left images show the underlying patterns of *fail* probabilities in their respective intensity $1 - \exp(-W_{ik}H_{k*})$.

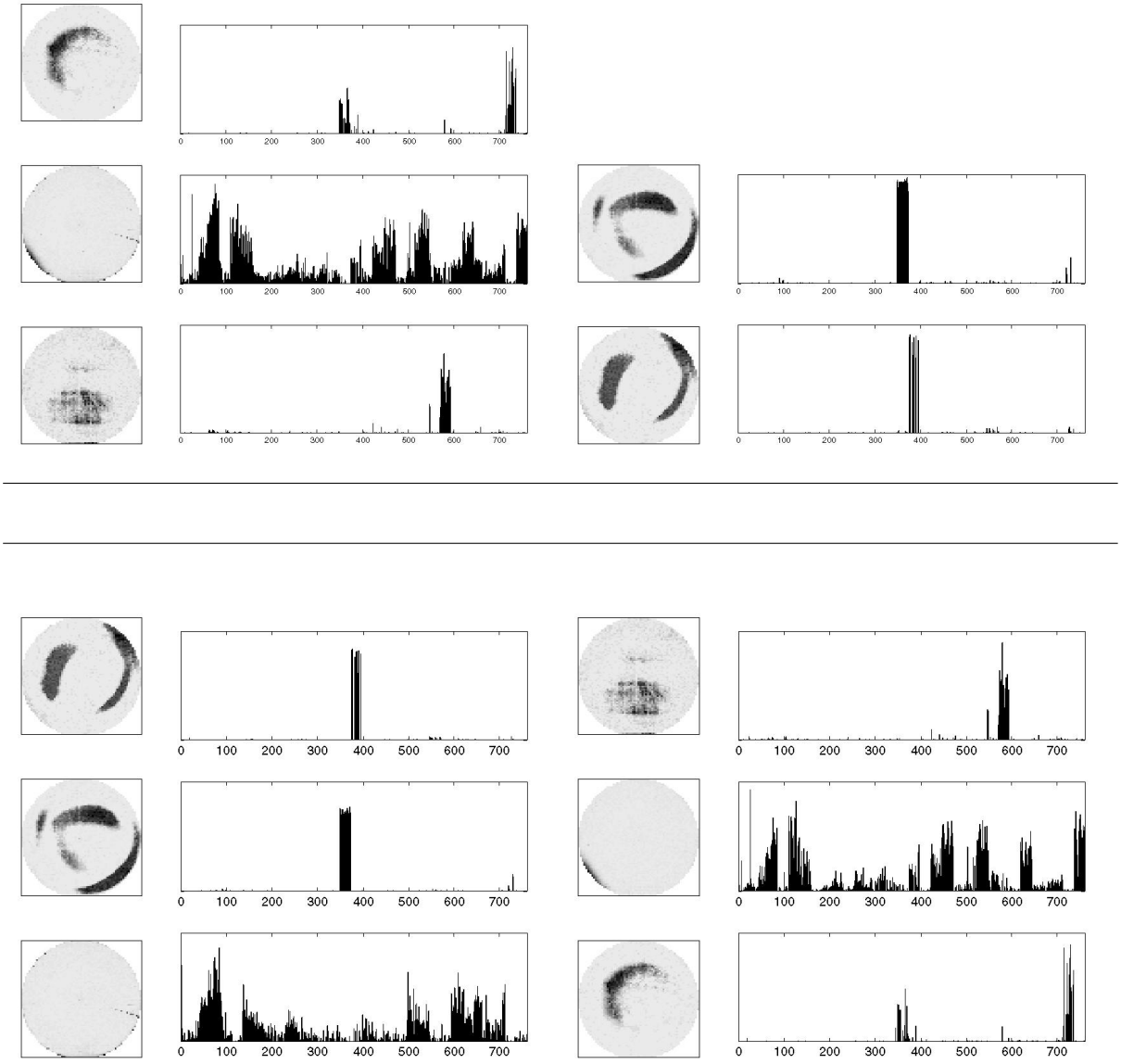

Figure 5.12: Real world example II: dateset comprising $N = 760$ wafers and $M = 3500$ chips
top: Decomposition in $K = 5$ sources (*left*: 1,2,3, *right*: 4,5) *bottom*: decomposition in $K = 6$ sources (*left*: 1,2,3, *right*: 4,5,6).
Visual inspection shows that four of the basis patterns $\mathbf{H}_{k*}$ are conserved while pattern $\mathbf{H}_{2*}^{K=5}$ is split into two new patterns $\mathbf{H}_{3*}^{K=6}$ and $\mathbf{H}_{5*}^{K=6}$. An interesting detail of this example are the peaks concerning bundles of 25 wafers representing the processing unit of a lot (e.g. in $\mathbf{W}_{*4}^{K=5}$ and $\mathbf{W}_{*5}^{K=5}$). The plots indicate that the corresponding basis pattern is present on all wafers of one lot.

5.6 Distinction from existing models

This section provides a short subsumption of existing literature on related topics. We will see that none of the common techniques is suitable to find a satisfactory solution to the described problem although some decisions on subproblems (e.g. in favour of a Bernoulli likelihood as cost function) have been motivated by citations.
There are two prominent models which utilize the Bernoulli likelihood to model binary data via real-valued source and weight variables, namely the logistic PCA [Tip99], [SSU03] and the Aspect Bernoulli model [KB08]. We briefly discuss both of them and explain how our model is to be positioned right in between them. Further, we briefly discuss additional approaches to decompose binary datasets, noting that none of them covers all aspects of the *binNMF* problem setting (summarized in Fig.5.3).

5.6.1 Logistic PCA

Logistic PCA is an extension of usual principal component analysis (PCA) for dimensionality reduction of binary data. There are three main contributions on this topic which are very similar to each other, and we summarize them under the term *logistic PCA* ([Tip99], [CDS01], [SSU03]).
It is formulated as an optimization problem of the following likelihood which in our notation reads

$$P(X_{ij}|\mathbf{W},\mathbf{H}) = \sigma([WH]_{ij})^{X_{ij}}(1-\sigma([WH]_{ij}))^{1-X_{ij}} \tag{5.28}$$

where $\sigma(z) = (1+\exp(-z))^{-1}$ is the logistic function.
Tipping [Tip99] utilized the model as a method to visualize binary datasets. He used a variational technique to determine the optimal parameters.
Collins et al. [CDS01] extended PCA to the exponential family to provide dimensionality reduction schemes for non-Gaussian distributions and describe a general optimization scheme in terms of Bregman distances. Since the Bernoulli distribution is a member of the exponential family, the framework can also be applied to binary datasets and leads to the model given by eq. (5.28).
Schein et al. [SSU03] use a method closely related to the one in [Tip99], naming it logistic PCA and demonstrate its power on data reconstruction on a variety of real world problems.
Since there is no non-negativity constraint on the factor matrices $\mathbf{W}$ and $\mathbf{H}$ negative values will occur in the optimal solutions. These are not interpretable in the context of modeling a strictly additive superposition of non-negative source variables.
Logistic PCA is a nonlinear version of PCA and thus has inherited most of its advantages and drawbacks. While being very powerful when reconstructing high dimensional data using only a few components, it might be hard to interpret the meaning of extracted components themselves. This issue clearly has analogies to the continuous case, where non-negative matrix factorization ([LS99],[PT94]) leads to better interpretable results than linear PCA in applications where the non-negative superposition principle holds.
Logistic PCA and *binNMF* both optimize a Bernoulli likelihood which is parameterized by the set of Bernoulli parameters $\Theta_{ij} \in [0,1]$. Both methods utilize a non-linear function which transfers the matrix product $\mathbf{WH}$ onto the range of the Θ_{ij}. The main difference is given by the non-negativity constraints in our model.

5.6.2 Aspect Bernoulli

The Aspect Bernoulli model, which was recently introduced by Kabán et al. [KBH04], and Bingham et al. [BKF09], does not need a nonlinearity and decomposes the mean of the Bernoulli distribution $P(X_{ij}=0)_{AB} = \mathbf{WH}_{ij}$ instead of $P(X_{ij}=0) = 1-\exp(-\mathbf{WH}_{ij})$, which leads to the likelihood

$$P(X_{ij}|\mathbf{W},\mathbf{H}) = [WH]_{ij}^{X_{ij}}(1-[WH]_{ij})^{1-X_{ij}}. \tag{5.29}$$

In the Aspect Bernoulli model, both parameter matrices $\mathbf{W}$ and $\mathbf{H}$ are constrained to have entries in $[0, 1]$. The matrix entry corresponding to entry H_{kj} of our pattern matrix $\mathbf{H}$ stores the Bernoulli probability of the j^{th} attribute being *on*, conditioned on the latent aspect k, while the parameter corresponding to our weight W_{ik} is the probability of choosing a latent aspect k in observation i. In order to keep the product $[\mathbf{WH}]_{ij} \leq 1$, the entries of row $\mathbf{W}_{i*}$ are further restricted to sum to 1, which is a standard mixture model assumption. The latter is the key feature which renders the Aspect Bernoulli model not applicable for our purposes. Cases where more than one source is highly active at the same time cannot be displayed properly. Suppose there are two aspects or elementary causes highly active on one observation and each single cause has a probability larger than 0.5. This would yield $W_{i1}H_{1j} > 0.5$ and $W_{i2}H_{2j} > 0.5 \Rightarrow [\mathbf{WH}]_{ij} > 1$, contradicting $[\mathbf{WH}]_{ij} \in [0, 1]$. Hence, such cases cannot be handled by the Aspect Bernoulli model.
The authors of [BKF09] present a rather informative overview of related models and even mention that an intermediate model between logistic PCA and aspect Bernoulli could be constructed. The *binNMF* model is an example of such an intermediate model which is to be positioned right between logistic PCA and Aspect Bernoulli.

5.6.3 Noisy-OR models

Saund's multiple cause model [Sau95] discusses two neural network approaches with one hidden layer and an output/input layer, namely the WRITE-BLACK and the WRITE-WHITE-AND-BLACK model. The former describes a voting scheme in which hidden units either abstain (0) or vote ON (1). One single vote ON suffices to turn the prediction unit ON. This voting scheme is analogous to our idea of failure generation, except that we allow non-negative single probabilities, while the votes of the hidden units are binary.
Dayan and Zemel [DZ95] introduce competition into the generative model by forcing at most one cause to take responsibility on any occasion.
Recently, [SH06] modeled high-dimensional binary data as generated by a small number of hidden binary sources combined via noisy-OR units. Indeed, Singliar et al. obtain a conditional probability of the data given the latent source variables and coefficients which is identical to our log-likelihood model 5.9, except that the source variables are restricted to be exclusively binary. The authors suggest a variational EM algorithm to do the optimization. In each sub step, a set of variational parameters is introduced which leads to a lower bound to the actual log-likelihood. This bound is maximized with respect to the variational parameters in the E-step, and maximized with respect to the usual variables in the M-step. Unfortunately, neither of the described steps has a closed form solution and has to be solved via numerical methods. It is also unclear how close an optimum likelihood solution can be reached by the variational method, since only a lower bound is maximized in each step, and not the likelihood directly.
An extension of the method allowing non-negative source values could be an alternative to our gradient algorithm as a refinement procedure, since the *binNMF* model can best be interpreted as an extension to the noisy-OR model with nonnegative hidden units and connections, provided with the activation function $1 - \exp(-z)$ to generate the output.
However, even exact EM algorithms are greedy algorithms and hence suffer from local maxima. We expect our preprocessing procedure to search the parameter space for good initial points to increase the performance of the noisy-OR methodology, too.

5.6.4 Probabilistic latent semantic analysis

Probabilistic latent semantic analysis (PLSA) [Hof99], assumes that a document can be described by a generative process, in which words are related to topics and several topics can generate a document

(see paragraph 2.5.3). In our case, a topic corresponds to a hidden elementary cause, a document is analogous to a wafer, and a word resembles a fail chip.
The probability that word j occurs in document i can be expressed by a convex combination [Hof01]

$$P(d_i, w_j) = P(d_i) \sum_{k=1}^{K} P(w_j|s_k)P(s_k|d_i) \tag{5.30}$$

Similar to the Aspect Bernoulli model discussed in section (5.6.2), this is a linear model for the superposition, where the convexity constraint is imposed thereby keeping the composite expression between 0 and 1. This property excludes the case where several elementary causes are simultaneously active to a high degree, and hence this mixture assumption does not hold in the binary wafer test data scenario considered here. On the other hand, there is no such limiting mixture assumption in the *binNMF* model, and a non-competing combination of all possible sources is responsible for each wafer instead of a mixture distribution.
In [SH06], the differences between noisy-or component analysis (NOCA) and mixture-based approaches such as the PLSA model are discussed in a text analysis context. These differences directly transfer to the *binNMF* setting. NOCA and *binNMF* share a similar structure except for the allowed values of the two factor matrices.
As pointed out by [Hof99], [SRS08], [BJ06] and others, probabilistic latent variable models have strong connections to NMF and are identical in special cases. They differ from the current *binNMF* model beneath others in being strictly linear.

5.6.5 Other approaches

Latent Dirichlet Allocation (LDA) [BNJ03] is a Bayesian extension to PLSA which adds a parameterized distribution for the documents, but still retains the mixture model property. Buntine and Jakulin [BJ06] present a general framework called *Discrete Component Analysis* which generalizes some of the methods mentioned here and places them into context. Zhang et al. [ZDLZ07] apply the NMF methodology to decompose a binary matrix $\mathbf{X}$ into binary matrices $\mathbf{W}$ and $\mathbf{H}$ by enforcing solutions with binary factor matrices. The method is called *binary matrix factorization*. Meeds et al. [MGNR06] also introduce a model called binary matrix factorization in which a real-valued data matrix $\mathbf{X}$ is factorized into a triple $\mathbf{UWV^T}$, where $\mathbf{U}$ and $\mathbf{V}$ are binary and $\mathbf{W}$ is a real-valued matrix.

5.7 Summary

This chapter introduced a novel blind source separation approach called *binNMF* to extract systematic failure causes from binary wafer test data. A model for the data generating process based on a minimum of assumptions was discussed which is closely related to the well-known NMF procedure and can be interpreted as a binary extension thereof. A two stage optimization procedure was presented. In a preprocessing step, the space of possible solutions is explored by a fast NMF algorithm on a simplified approximation to the original problem. An alternating gradient procedure and a multiplicative algorithm were discussed to optimize the actual problem in form of a Bernoulli log-likelihood in a second fine-tuning step. The model was placed into context of existing literature on related topics, concluding that none of the existing techniques provide a satisfactory solution to the described problem. The performance of the overall procedure was illustrated in an artificial toydata example. Two real world applications demonstrate the potential of the *binNMF* technique to extract meaningful components from binary real world datasets. While the usual uniqueness problem of NMF was shown

to be irrelevant in this binary extension, the optimal number of sources remains an open question here.
In the rest of this thesis, we will investigate Bayesian techniques and their potential to determine an optimal number of components automatically.

Chapter 6

Bayesian learning

Bayesian techniques are statistical approaches to data modeling which offer a structured way to incorporate uncertainty.
Bayesian inference means techniques in which observations are used to compute a model which is a joint probability distribution of observations, variables and parameters.
The most important equation in Bayesian inference is given by Bayes' theorem

$$P(X,Y) = P(X)P(X|Y) = P(Y)P(Y|X) \tag{6.1}$$

which states that the joint probability distribution function of two events or hypotheses X and Y can be expressed in terms of *marginal* and *conditional* distributions.
If event X depends on an unknown variable Z, we can express all available information on Z by some prior distribution $P(Z)$ and integrate the variable out by the use of a second fundamental equation:

$$P(X) = \int P(X,Z)dZ = \int P(X|Z)P(Z)dZ \tag{6.2}$$

which integrates over all possible values of Z.

6.1 Bayesian inference

The Bayesian probability of an event X is a person's *degree of belief* in that event [Hec98]. Whereas a classical probability is a physical property of the world, a Bayesian probability is a property of the person who assigns the probability. An important difference between both, classical and Bayesian probability is that one need not undertake repeated trials to determine a Bayesian probability.
Ghahramani [Gha04] explains the basic idea of Bayesian learning as follows: Suppose we wish to design a machine which is able to explore the world all by itself. The machine has beliefs about the world, and updates these beliefs on the basis of observed data. We must equip this machine with some basic methodology to do this. It can be shown that, if certain axioms of coherent inference are accepted, and if the machine represents the strength of its beliefs by real numbers, then the only reasonable way of manipulating these beliefs is to follow the rules of probability [Jay03].
The machine permanently tries to explain the world by inventing new models and collecting data. MacKay [Mac92a] distinguishes between two levels of Bayesian inference: At the beginning, data $\mathbf{X}$ are gathered and alternative models are created. In the first stage of inference, the parameters of each model are adjusted to the data. In the second stage of inference, we assign preferences to the alternative models. On this basis, new data is gathered, new models are invented or future actions are chosen.

Let a model m^i be parameterized by the set of parameters $\mathbf{\Theta}^i$.

1. For every single model m^i, the posterior probability of its parameters $\mathbf{\Theta}^i$ is computed according to
$$P(\mathbf{\Theta}^i|\mathbf{X}, m^i) = \frac{P(\mathbf{X}|\mathbf{\Theta}^i, m^i)P(\mathbf{\Theta}^i|m^i)}{P(\mathbf{X}|m^i)} \tag{6.3}$$
In the first stage of inference, the parameters of a fixed model are adjusted to the data by computing their posterior distribution.

2. At the second level of inference, the posterior probability of every model is computed according to
$$P(m^i|\mathbf{X}) = \frac{P(\mathbf{X}|m^i)P(m^i)}{P(\mathbf{X})} \tag{6.4}$$
Based on the models' posterior probabilities, we assign preferences to the alternative models.

Note that the data dependent term in the second stage on the right hand side of eq. (6.4) is the normalizing constant in the fist stage

$$P(\mathbf{X}|m^i) = \int P(\mathbf{X}|\mathbf{\Theta}^i, m^i)P(\mathbf{\Theta}^i|m^i)d\mathbf{\Theta}^i \tag{6.5}$$

Assuming that there is no reason to assign different priors $P(m^i)$ and $P(m^j)$ to alternative models i and j, different models are ranked by evaluating the *evidence* $P(\mathbf{X}|m^i)$ which can be interpreted as the support the data gives us for preferring model m^i.
Except for very simple models, this integral is usually very complex and not computable directly.
In the Bayesian statistics community *Markov Chain Monte Carlo* (MCMC) is the method of choice for approximating difficult high dimensional expectations and integrals [BG04]. Several methods have been proposed for estimating the evidence, such as Chib's method [Chi95], or Annealed Importance Sampling [Nea01].
Sampling methods are prohibitive for large-scale problems, because they can be slow and the posterior distribution over parameters is stored as a set of samples and can yield problems concerning memory.
Another approach is *Laplace's method* which makes a local Gaussian approximation to the *a posteriori* distribution around its maximum (the MAP estimate) [Mac03]. In the large sample limit, this approximation becomes more and more accurate, given some regularity conditions [BG04].
Variational methods [JGJS98] are deterministic approximations which make use of a lower bound by suitable choices of a variational distribution and will be discussed in more detail in section 6.2.

6.1.1 Statistical physics

The techniques used to approximate the necessary integrals mentioned above also find application in statistical physics, for example to evaluate the partition function of a large system of interacting spins. Each state of N spins can be identified with a probability distribution

$$P(\mathbf{x}|\beta, \mathbf{J}) = \frac{1}{Z(\beta, \mathbf{J})} \exp[-\beta E(\mathbf{x}; \mathbf{J})] \tag{6.6}$$

where $\mathbf{x}$ is a state vector $\mathbf{x} \in \{-1, +1\}^N$ and $E(\mathbf{x}; \mathbf{J})$ is the energy of the state. The parameters $\mathbf{J}$ are coupling constants and $\beta = \frac{1}{k_B T}$ with k_B denoting Boltzmann's constant and T a temperature.
The normalization constant is called a *partition function*

$$Z(\beta, \mathbf{J}) = \sum_{\mathbf{x}} \exp[-\beta E(\mathbf{x}; \mathbf{J})] \tag{6.7}$$

where the summation goes over all possible states of $\mathbf{x}$.
Many important statistical quantities of the system can be derived from the partition function, such as its energy $\langle E \rangle = \frac{\partial ln Z}{\partial \beta}$, heat capacity $C_v = \frac{\partial \langle E \rangle}{\partial T}$ or entropy $S = \frac{\partial}{\partial T}(k_B T \ln Z)$.

6.1.2 Graphical models

Probabilistic models for which a graph denotes the conditional independence structure between random variables are called graphical models. Probabilistic inference in graphical models means the computation of a conditional probability distribution over the values of some hidden variables $\mathbf{Y}$, given the values of some observed variables $\mathbf{X}$

$$P(\mathbf{Y}|\mathbf{X}) = \frac{P(\mathbf{Y}, \mathbf{X})}{P(\mathbf{X})} \tag{6.8}$$

There are directed graphs and undirected graphs. Bayesian networks are directed graphs [Hec98] which encode the joint probability distribution of variables. Each variable is expressed as a node in the graph and depends on its parent nodes.
An undirected graphical model or *Markov random field* [JGJS98] can be specified by associating *potentials* with *cliques* in the graph. A clique is defined as a maximal subset of fully connected nodes. A potential is a function of the set of configurations of a clique that associates a positive real number with each configuration. For every subset of nodes C_i that forms a clique, there is an associated potential $\phi_i(C_i)$. The joint probability distribution for all nodes in the graph S is the product over the clique potentials

$$P(S) = \frac{\prod_i \phi_i(C_i)}{Z} \tag{6.9}$$

where the sum over all possible configurations

$$Z = \sum_{\{S\}} \prod_i \phi_i(C_i) \tag{6.10}$$

just like in statistical mechanics, is called a *partition function.*
Directed graphical models can be transformed into undirected graphical models.
Many prominent examples can be represented as graphical models: Neural networks [SJ98] are layered graphs with a nonlinear *activation function* at each node. Boltzmann machines [AHS85], [HS86] are undirected graphical models with binary-valued nodes and a restricted set of potential functions.

6.1.3 Bayesian parameter estimation

Bayesian techniques offer a principled way to combine prior knowledge with data and motivate a coherent framework for machine learning. Bayesian parameter estimation is stage 1 in the above inference procedure described by eq. (6.3).

Maximum a posteriori estimation

Let the data $\mathbf{X}$ be generated by a model with parameters $\mathbf{\Theta}$. Note that here the conditioning on a fixed model m^i is assumed implicitly. Further let $P(\mathbf{\Theta})$ denote our initial (prior) belief on the parameters' values. In case of a vague guess, this can e.g. be a Gaussian around some expected value. In absence of any opinion on the parameters, this can be some flat distribution, e.g. a uniform.
According to Bayes' formula $\mathbf{X}$.

$$\underbrace{P(\mathbf{X}|\mathbf{\Theta})}_{likelihood}\underbrace{P(\mathbf{\Theta})}_{prior} = \underbrace{P(\mathbf{\Theta}|\mathbf{X})}_{posterior}\underbrace{P(\mathbf{X})}_{evidence} \quad = \quad \underbrace{P(\mathbf{X}, \mathbf{\Theta})}_{joint\ distribution} \tag{6.11}$$

the relevant quantities can be transformed into each other.

- The *prior* distribution $P(\mathbf{\Theta})$ expresses our knowledge on the parameters $\mathbf{\Theta}$ before seeing the data $\mathbf{X}$
- The *posterior* distribution $P(\mathbf{\Theta}|\mathbf{X})$ reflects our knowledge on $\mathbf{\Theta}$ given $\mathbf{X}$
- The *likelihood* $P(\mathbf{X}|\mathbf{\Theta})$ is a function of the parameters $\mathbf{\Theta}$ and describes the probability to observe the data $\mathbf{X}$ if the parameters are set to $\mathbf{\Theta}$.
- The *evidence* $P(\mathbf{X}) = \int P(\mathbf{X}|\mathbf{\Theta})P(\mathbf{\Theta})d\mathbf{\Theta}$ is the expected likelihood under the prior distribution, and indicates how much the observed data $\mathbf{X}$ supports the model assumption (e.g. the structure of the parameters $\mathbf{\Theta}$ and the likelihood function).

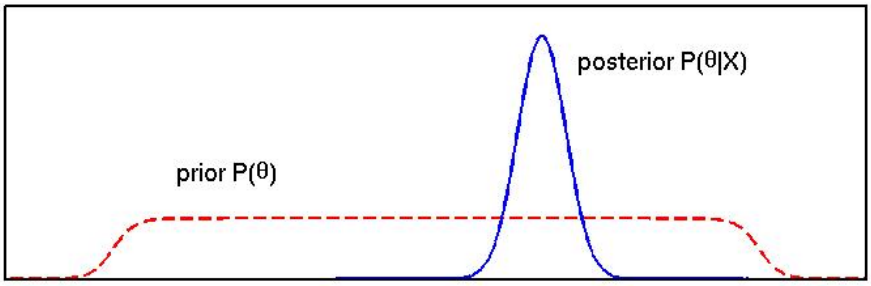

Figure 6.1: MAP estimation: The arrival of the data $\mathbf{X}$ turns our prior belief on the parameters $P(\mathbf{\Theta})$ into a posterior distribution $P(\mathbf{\Theta}|\mathbf{X})$ via Bayes' formula

So, the most probable parameters $\mathbf{\Theta}$ in the light of the data can be computed by maximizing the quantity

$$P(\mathbf{\Theta}|\mathbf{X}) \propto P(\mathbf{X}|\mathbf{\Theta})P(\mathbf{\Theta}) \tag{6.12}$$

since $P(\mathbf{X})$ does not depend on particular parameters.
This is called maximum a posteriori (MAP) estimation and extends the maximum likelihood (ML) methodology by offering the possibility to incorporate prior knowledge on the parameters.
Usually it is computationally convenient to search for the maximum of the logarithm

$$\log\left(P(\mathbf{X}|\mathbf{\Theta})P(\mathbf{\Theta})\right) = \log(P(\mathbf{X}|\mathbf{\Theta})) + \log(P(\mathbf{\Theta})) \tag{6.13}$$

instead.

Example: Neural network's constraint for large weight parameters.

The weight decay parameter in neural networks can be derived by a prior distribution on the parameters (here the couplings between the neurons)

$$P(\mathbf{\Theta}) \propto \exp\left(-\beta\Theta^2\right) \tag{6.14}$$

which is a Gaussian distribution with mean 0 and $\beta = \frac{1}{2\sigma^2}$.
The MAP estimate is the argument of the maximum of the quantity

$$\log(P(\mathbf{X}|\mathbf{\Theta})) - \beta\Theta^2 \tag{6.15}$$

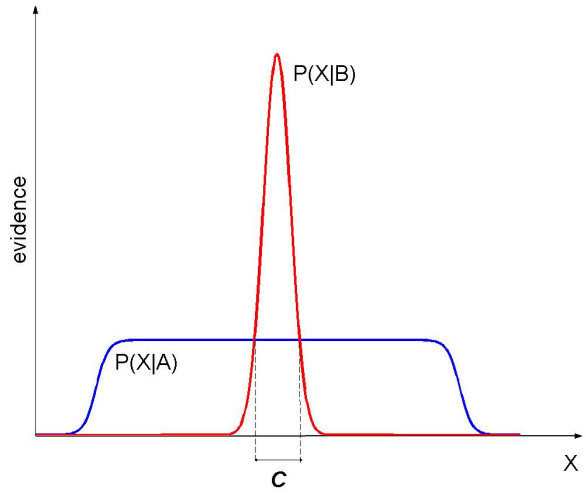

Figure 6.2: Why Bayesian inference embodies Occam's razor
The horizontal axis represents the space of all possible datasets $\mathbf{X}$. A simple model B with few adjustable parameters makes only a limited range of predictions, shown by $P(\mathbf{X}|B)$, while a more powerful model A which has more free parameters is able to predict a greater variety of datasets. If the observed data comes from region C, the evidence for the simpler model B is greater than the evidence for model A. (figure and explanation are borrowed from [Mac03])

given some likelihood function $P(\mathbf{X}|\mathbf{\Theta})$ (e.g. the quadratic deviation).
Summarizing, the MAP framework is the statistical formulation of certain optimization problems which can be written as a cost function with regularization.
The negative log likelihood corresponds to the error function, while the regularizer can be interpreted in terms of a log prior distribution over the parameters.

Conjugate priors

For a given likelihood function $P(\mathbf{X}|\mathbf{\Theta})$ it is sometimes possible to choose the prior $P(\mathbf{\Theta})$ such that the posterior $P(\mathbf{\Theta}|\mathbf{X})$ has the same functional form as the prior.
The prior is said to be *conjugate* to the likelihood then [RS61].
If the likelihood belongs to the exponential family of distributions, there is always such a conjugate prior.

6.1.4 Bayesian model selection and Occam's razor

It is always possible to better explain a given dataset by a model with more parameters. Think of a curve fitting problem by a set of splines. A given noisy curve can fitted arbitrarily well if the number of splines used for the fit is increased without limitation. If the number of splines is too large, however, the procedure will start to fit the noise and not the curve of interest. This is known as *overlearning* or *over-fitting* [Bis96].
The Bayesian procedure for model comparison (stage 2) automatically incorporates a principle called *Occam's razor*, which states that *"one should not increase, beyond what is necessary, the number of entities required to explain anything"* [citation found online at http://pespmc1.vub.ac.be/occamraz.html].

The evidence $P(\mathbf{X}|A)$ that the data gives us for model A is a transportable quantity which allows the comparison of completely different model structures with each other.

6.1.5 Examples for the evidence framework

The evidence framework has been applied e.g. in neural networks or interpolation tasks:

Neural Networks

MacKay [Mac92b] applied Bayesian inference techniques to the determination of optimal parameters which change the effective learning model, e.g. the number of hidden units or weight decay terms in backpropagation networks. A set of input-target pairs $D = \{\mathbf{x}^m, \mathbf{t}^m\}$ is used to train a network with architecture $\mathcal{A}$ and connections $\mathbf{w}$ to make predictions on the target outputs $\mathbf{t}^m$ as a function of input $\mathbf{x}^m$ in accordance with the probability function $P(\mathbf{t}^m|\mathbf{x}^m, \mathbf{w}, \beta, \mathcal{A})$. A prior probability $P(\mathbf{w}|\alpha, \mathcal{A})$ is assigned to alternative network connection strengths $\mathbf{w}$, and a posterior probability $P(\mathbf{w}|D, \alpha, \beta, \mathcal{A})$ of the network connections $\mathbf{w}$ is derived which depends on a decay rate α, the data error and a noise parameter β. Different model structures can be compared by evaluating their evidence $P(\mathcal{A}|D)$ which involves the determination of optimal values for the parameters α, β and the weights $\mathbf{w}$ (see [Mac92b], [Mac99] for details).

Interpolation

The same framework of Bayesian model comparison was also applied by MacKay in a curve fitting context as *Bayesian Interpolation* [Mac92a]. The interpolation examples include the determination of an optimal number of basis functions in a polynomial model, to set the characteristic size in a radial basis function (rbf) model and to determine the regularizer in a spline model. It was demonstrated that the evidence framework can also be applied to compare the performance of these three totally different models in the light of the data.

6.2 Variational Bayes

Variational techniques are used as approximation methods in a variety of fields such as statistics [Rus76], statistical mechanics and quantum statistics [Fey72], [Par88], quantum mechanics [Mit94], or finite element analysis [Bat96]. The strategy is always similar: a complex problem is converted into a simpler one by decoupling the degrees of freedom in the original problem. This decoupling is achieved by an expansion which includes additional adjustable parameters.
A famous example for variational techniques in physics are *mean field* approximations [Mac03], where e.g. the interactions of many spins are replaced by an averaged quantity or mean field to make the computations tractable.

6.2.1 A lower bound for the log evidence

In general, variational approximations are deterministic procedures which provide bounds for probabilities of interest.
One possible bound is induced by Jensen's inequality ([Jen06]). Let $\mathbf{X}$ denote the observed variables, $\mathbf{Y}$ denote the latent variables, and $\boldsymbol{\Theta}$ denote the parameters of a fixed model m which will be implicitly

assumed in the following. The log evidence can be bounded from below

$$\begin{aligned}
\ln P(\mathbf{X}) &= \ln \int P(\mathbf{X},\mathbf{Y},\mathbf{\Theta}) d\mathbf{Y} d\mathbf{\Theta} && (6.16)\\
&= \ln \int Q(\mathbf{Y},\mathbf{\Theta}) \frac{P(\mathbf{X},\mathbf{Y},\mathbf{\Theta})}{Q(\mathbf{Y},\mathbf{\Theta})} d\mathbf{Y} d\mathbf{\Theta} && (6.17)\\
&\geq \int Q(\mathbf{Y},\mathbf{\Theta}) \ln \frac{P(\mathbf{X},\mathbf{Y},\mathbf{\Theta})}{Q(\mathbf{Y},\mathbf{\Theta})} d\mathbf{Y} d\mathbf{\Theta} && (6.18)\\
&= \int Q(\mathbf{Y},\mathbf{\Theta}) \ln \frac{P(\mathbf{Y},\mathbf{\Theta}|\mathbf{X})P(\mathbf{X})}{Q(\mathbf{Y},\mathbf{\Theta})} d\mathbf{Y} d\mathbf{\Theta} && (6.19)\\
&= \int Q(\mathbf{Y},\mathbf{\Theta}) \ln \frac{P(\mathbf{Y},\mathbf{\Theta}|\mathbf{X})}{Q(\mathbf{\Theta})} d\mathbf{Y} d\mathbf{\Theta} + \int Q(\mathbf{Y},\mathbf{\Theta}) \ln P(\mathbf{X}) d\mathbf{Y} d\mathbf{\Theta} && (6.20)\\
&= -KL(Q(\mathbf{Y},\mathbf{\Theta})||P(\mathbf{Y},\mathbf{\Theta}|\mathbf{X})) + \ln P(\mathbf{X}) && (6.21)
\end{aligned}$$

The first expression in the last line is the negative Kullback-Leibler divergence between the two probability distributions $Q(\mathbf{\Theta})$ and $P(\mathbf{\Theta}|\mathbf{X})$.
According to Gibb's inequality, the KL-divergence between two probability distributions Q and P is non-negative always and zero only if $Q = P$. This is a well-known result from statistical physics and information theory [CT91], [Mac03]. Hence, maximizing the latter quantity w.r.t. the variational distribution $Q(\mathbf{Y},\mathbf{\Theta})$ leads to

$$Q(\mathbf{Y},\mathbf{\Theta}) = P(\mathbf{Y},\mathbf{\Theta}|\mathbf{X}) \tag{6.22}$$

this means that the variational distribution equals the posterior distribution of the latent variables and parameters.
Usually, the computation of the true posterior $P(\mathbf{Y},\mathbf{\Theta}|\mathbf{X})$ is intractable.
Hence, simpler forms of the approximating distributions are chosen which render the problem tractable. Hinton and van Camp [HvC93], Hinton and Zemel [HZ94] use a separable Gaussian form for Q in a similar task.
In contrast, the so called free-form approach [Att00],[GB00b], assumes the following factorized approximation

$$Q(\mathbf{Y},\mathbf{\Theta}) \approx Q(\mathbf{Y})Q(\mathbf{\Theta}) \tag{6.23}$$

which yields a lower bound

$$\ln P(\mathbf{X}) \geq \int Q(\mathbf{Y})Q(\mathbf{\Theta}) \ln \frac{P(\mathbf{X},\mathbf{Y},\mathbf{\Theta})}{Q(\mathbf{Y})Q(\mathbf{\Theta})} d\mathbf{Y} d\mathbf{\Theta} =: F_Q \tag{6.24}$$

In general, this factorized approximation of the true posterior will not reach equality and remain a lower bound on the log evidence. The strategy of variational Bayes is thus to make this lower bound as tight as possible.
It is interesting to rewrite the bound in eq.(6.24) as

$$\begin{aligned}
F_Q &= \int Q(\mathbf{Y})Q(\mathbf{\Theta}) \ln \frac{P(\mathbf{X},\mathbf{Y}|\mathbf{\Theta})P(\mathbf{\Theta})}{Q(\mathbf{Y})Q(\mathbf{\Theta})} d\mathbf{Y} d\mathbf{\Theta} && (6.25)\\
&= \int Q(\mathbf{Y})Q(\mathbf{\Theta}) \ln \frac{P(\mathbf{X},\mathbf{Y}|\mathbf{\Theta})}{Q(\mathbf{Y})} d\mathbf{Y} d\mathbf{\Theta} + \int Q(\mathbf{\Theta}) \ln \frac{P(\mathbf{\Theta})}{Q(\mathbf{\Theta})} d\mathbf{\Theta} && (6.26)\\
&= \left\langle \ln \frac{P(\mathbf{X},\mathbf{Y}|\mathbf{\Theta})}{Q(\mathbf{Y})} \right\rangle_{Q(\mathbf{Y})Q(\mathbf{\Theta})} - KL(Q(\mathbf{\Theta})||P(\mathbf{\Theta})) && (6.27)
\end{aligned}$$

where the first term corresponds to the averaged log likelihood and the second term is the KL distance between the prior and posterior approximation of the parameters. With increasing number of parameters $|\mathbf{\Theta}|$, the KL distance grows and F_Q decreases. Hence, a large number of parameters is automatically penalized by the variational Bayes framework. This was pointed out by Attias [Att00], who also showed that the popular Bayesian information criterion (BIC) for model order selection [Sch78] is a limiting case of the variational Bayes framework. In the large sample limit $N \to \infty$, the parameter posterior is sharply peaked about the most probable value $\mathbf{\Theta} = \mathbf{\Theta}^*$ and the KL-term in F_Q reduces to $\frac{|\mathbf{\Theta}^*|}{2} \ln N$.

6.2.2 The VBEM algorithm

The Variational Bayesian EM algorithm [Att00], [GB01], [BG04] iteratively maximizes F_Q in eq. 6.24 w.r.t. the free distributions $Q(\mathbf{Y})$ and $Q(\mathbf{\Theta})$. Each update is done while keeping the other quantities fixed.
Setting

$$\frac{\partial}{\partial \mathbf{Q}(\mathbf{\Theta})} F_Q + \lambda \left(\int Q(\mathbf{\Theta}) d\mathbf{\Theta} - 1 \right) = 0 \tag{6.28}$$

where λ is a Lagrange multiplier ensuring the normalization of the density Q immediately leads to

$$Q(\mathbf{\Theta}) \propto P(\mathbf{\Theta}) \exp \left[\int \ln P(\mathbf{X}, \mathbf{Y}|\mathbf{\Theta}) Q(\mathbf{X}) d\mathbf{X} \right] \tag{6.29}$$

Similarly, one can derive

$$Q(\mathbf{X}) \propto \exp \left[\int \ln P(\mathbf{X}, \mathbf{Y}|\mathbf{\Theta}) Q(\mathbf{\Theta}) d\mathbf{\Theta} \right] \tag{6.30}$$

The equations (6.30) and (6.29) constitute the Variational Bayesian EM algorithm. Beal and Ghahramani [BG04] show that the Variational Bayesian EM algorithm reduces to the ordinary EM algorithm if the parameter density is restricted to be a Dirac delta function $Q(\mathbf{\Theta}) = \delta(\mathbf{\Theta} - \mathbf{\Theta}^*)$.
Attias [Att99], [Att00] was the first who described the variational Bayes framework and showed that it is a generalization of the well-known EM algorithm [DLR77] which is the method of choice for maximum likelihood parameter estimation in statistics. Before, Neal and Hinton [NH98] generalized the EM algorithm for maximum likelihood estimation to cases where a lower bound is iteratively maximized w.r.t. parameters and hidden variables.
Ghahramani and Beal [GB01] applied it to the large class of conjugate-exponential models, which can be characterized by two conditions:

1. The complete-data likelihood is in the exponential family, i.e. can be written as

$$P(\mathbf{X}, \mathbf{Y}|\mathbf{\Theta}) = g(\mathbf{\Theta}) f(\mathbf{X}, \mathbf{Y}) e^{\mathbf{\Phi}^T \mathbf{u}(\mathbf{X}, \mathbf{Y})} \tag{6.31}$$

   where $\mathbf{\Phi}(\mathbf{\Theta})$ is the vector of natural parameters, $\mathbf{u}$ and f are functions and g is a normalization constant.

2. The parameter prior is conjugate to the complete-data likelihood

$$P(\mathbf{\Theta}|\eta, \nu) = h(\eta, \nu) g(\mathbf{\Theta})^\eta e^{\mathbf{\Phi}^T \nu} \tag{6.32}$$

   where η and ν are hyperparameters of the prior and h is a normalization constant.

6.2.3 Applications of variational inference

Special attention has been spent on variational methods in context of graphical models (see [JGJS98] for a tutorial on variational methods in graphical models). In densely connected graphical models there are often averaging phenomena which render nodes relatively insensitive to particular values of their neighbors. Variational methods take advantage of these averaging phenomena and can lead to simple approximation procedures.
Machine learning applications of variational Bayes include Ensemble learning for neural networks [HvC93] and mixtures of experts [WM96],
Utilizing mixture models as approximating distributions are discussed in [JJ98], [BLJJ98] and [Att00].
Variational Bayes versions of popular data analysis techniques have been developed, such as Bayesian logistic regression [JJ97], Bayes mixtures of factor analyzers [GB00b], Variational Principal Component Analysis [Bis99], Ensemble learning for Independent Component Analysis [Lap99], Factor analysis [HK07], or Bayesian Independent Component Analysis [WP07].

Statistical physics, continued

In statistical physics, variational free energy minimization [Mac03] is a method which approximates the (usually very complex) distribution $P(\mathbf{x}|\beta, \mathbf{J})$ given in eq. (6.6) by a simpler one $Q(\mathbf{x}; \theta)$ that is parameterized by adjustable parameters θ. The quality of the approximation can be measured by the *variational free energy*

$$\beta\tilde{F}(\theta) = \sum_{\mathbf{x}} Q(\mathbf{x}; \theta) \ln \frac{Q(\mathbf{x}; \theta)}{P(\mathbf{x}|\beta, \mathbf{J})} - \ln Z(\beta, \mathbf{J}) \tag{6.33}$$

which is the sum of the Kullback-Leibler divergence or *relative entropy* $D_{KL}(Q||P)$ between the two distributions Q and P and the true *free energy* of the system defined as

$$\beta F := -\ln Z(\beta, \mathbf{J}) \tag{6.34}$$

Thus, the variational free energy $\beta\tilde{F}(\theta)$ is bounded below by the true free energy βF and the two quantities are equal if $Q(\mathbf{x}; \theta) = P(\mathbf{x}|\beta, \mathbf{J})$.
The optimization strategy is to vary the parameters θ such as to minimize $\beta\tilde{F}(\theta)$. The approximating distribution Q is then a simplified approximation to the true distribution P, and the value of $\beta\tilde{F}(\theta)$ will be an upper bound to βF.

Chapter 7

Bayesian approaches to NMF

While we introduced the technique of NMF in chapter 2, and gave a general introduction on Bayesian learning theory in chapter 6, this chapter brings both approaches together. First, the statistical aspects of usual NMF are discussed in section 7.1. Then, section 7.2 briefly reviews existing literature on Bayesian approaches to NMF.

7.1 The statistical perspective of NMF

While in chapter 2, NMF was formulated as an optimization problem of a cost function with non-negativity (and other optional) constraints, we will discuss the statistical perspective of NMF in this section.
We will recognize that plain NMF can be interpreted as maximum likelihood estimation with additional non-negativity constraints.
Suitable prior distributions can be used to enforce certain characteristics in the solutions and NMF with additional constraints can be interpreted as MAP estimation.
As an example, we show that non-negative sparse coding [Hoy02] can be derived from MAP estimation with an exponential prior.

7.1.1 NMF as Maximum likelihood estimation

The interpretation of NMF as maximum likelihood estimation has been recognized by several authors (see e.g. [SDB+03], [CZA06], [SL08], [FC09] and others).

Gaussian noise

Assuming the usual NMF model with additional Gaussian noise:

$$\mathbf{X} = \mathbf{WH} + \mathbf{E} \Leftrightarrow \mathbf{E} = \mathbf{X} - \mathbf{WH} \tag{7.1}$$

where the matrix $\mathbf{E}$ denotes the reconstruction error whose entries E_{ij} are assumed to be independent identical distributed according to a Gaussian with zero mean and variance σ_r^2 (r stands for *reconstruction error*). Due to the independence assumption the joint distribution of all data items factorizes:

$$P(\mathbf{X}|\mathbf{W},\mathbf{H}) = \left(-\frac{1}{\sqrt{2\pi}\sigma_r}\right)^{NM} \prod_i \prod_j \frac{1}{\sqrt{2\pi}\sigma_r} e^{\left(\frac{1}{2}\left(\frac{X_{ij}-[\mathbf{WH}]_{ij}}{\sigma_r}\right)^2\right)} \tag{7.2}$$

Taking the logarithm yields

$$\ln(P(\mathbf{X}|\mathbf{W},\mathbf{H})) = -NM\ln(\sqrt{2\pi}\sigma_r) - \underbrace{\frac{1}{2\sigma_r^2}\sum_i\sum_j (X_{ij} - [\mathbf{WH}]_{ij})^2}_{D_E(\mathbf{X},\mathbf{WH})} \tag{7.3}$$

Maximizing of the latter quantity w.r.t. $\mathbf{W},\mathbf{H}$ neglects the constant terms and we see immediately that the minimization of the squares Euclidean distance $D_E(\mathbf{X},\mathbf{WH})$ for NMF (see eq. 2.4 in chapter 2) is equivalent to the maximization of a Gaussian likelihood. Note that the non-negativity constraint on $\mathbf{W}$ and $\mathbf{H}$ must be stated in additionally.

Poisson noise

Poisson noise is in general used to describe the noise affecting counting processes, and is sometimes called *photon noise* [BLZ07]. The pdf of a Poisson distribution is given by

$$P(n;\lambda) = \frac{\lambda^n}{n!}\exp(-\lambda) \tag{7.4}$$

where n is an integer and λ is the expected number of occurrences during a certain time interval here. If we interpret non-negative numbers (such as pixel intensities in an image) as photon counts (e.g. by virtually multiplying each value by some very large number), we can slightly abuse the notation and assume that a pixel X_{ij} is generated by a Poisson process with parameter $[\mathbf{WH}]_{ij}$. If we further assume that all entries of $\mathbf{X}$ are independent of each other (the dependency structure is later induced by the matrix product), we can write

$$P(\mathbf{X}|\mathbf{W},\mathbf{H}) = \prod_i\prod_j \frac{[\mathbf{WH}]_{ij}^{X_{ij}}\exp(-[\mathbf{WH}]_{ij})}{X_{ij}!} \tag{7.5}$$

Taking the logarithm yields

$$\ln(P(\mathbf{X}|\mathbf{W},\mathbf{H})) = \sum_i\sum_j X_{ij}\ln[\mathbf{WH}]_{ij} - [\mathbf{WH}]_{ij} - \ln(X_{ij}!) \tag{7.6}$$

Stirling's formula ($\ln(n!) \approx n\ln(n) - n$ for $n >> 1$) leads to an approximation

$$\ln(P(\mathbf{X}|\mathbf{W},\mathbf{H})) \approx \sum_i\sum_j X_{ij}\ln\frac{[\mathbf{WH}]_{ij}}{X_{ij}} - [\mathbf{WH}]_{ij} + X_{ij} \tag{7.7}$$

which is exactly the negative of the generalized Kullback-Leibler divergence for NMF $D_{KL}(\mathbf{X},\mathbf{WH})$. Again, the non-negativity constraints must be stated explicitly.
The two preceding examples show that the two widespread plain cost functions evolve automatically in a maximum likelihood approach by assuming either additive Gaussian noise or Poisson noise.
However, the non-negativity constraint on the parameter matrices must be claimed in addition to the maximum likelihood procedure. Similarly, the assumption of multiplicative Gamma noise can be shown to lead to the Itakura-Saito divergence [FBD09].

7.1.2 Regularized NMF as MAP estimation

In paragraph 6.1.3, we introduced MAP estimation as maximization of the following quantity

$$P(\mathbf{\Theta}|\mathbf{X}) \propto P(\mathbf{X}|\mathbf{\Theta})P(\mathbf{\Theta}) \tag{7.8}$$

where $\mathbf{X}$ is the data and $\mathbf{\Theta}$ are the parameters of a fixed model.
In the NMF-setting, the parameters are given by the two factor matrices $\mathbf{\Theta} = \{\mathbf{W}, \mathbf{H}\}$. In a Bayesian setting, any prior knowledge must be incorporated by prior distributions. So for NMF, we need to formulate the non-negativity e.g. of $\mathbf{W}$ by a prior distribution which is nonzero only if every entry $W_{ik} \geq 0$.
There are lots of possible probability distributions on the non-negative reals, such as Exponential, Gamma, Beta and others. Moreover, a non-negative distribution can be easily derived from any distribution over the whole reals by truncating all negative elements and re-normalizing the remaining distribution such that is integrates to 1.
An example is the rectified Gaussian distribution is the rectified Gaussian distribution

$$\mathcal{N}^+(x|\mu,\sigma) =: \begin{cases} \frac{1}{Z_{\mathcal{N}^+}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, & 0 \leq x < \infty \\ 0 & \text{otherwise} \end{cases} \tag{7.9}$$

where the normalizing constant is given by

$$Z_{\mathcal{N}^+} = \frac{1}{2}\sigma\sqrt{2\pi}\ \text{erfc}\left(-\frac{\mu}{\sigma\sqrt{2}}\right)$$

and $\text{erfc}(z) =: \frac{2}{\sqrt{\pi}} \int_z^\infty \exp(-t^2)dt$ is the complementary error function.
The absence of additional knowledge on a parameter W_{ik} except its non-negativity could e.g. be encoded by

$$P(W_{ik}) = \begin{cases} C_W, & \text{if } 0 \leq W_{ik} \leq W_{max} \\ 0 & \text{else.} \end{cases} \tag{7.10}$$

where C_W and W_{max} are constants. Note that the actual value of W_{max} is arbitrary, but must be finite such that the distribution can be normalized $\int P(W_{ik})dW_{ik} = 1 \Rightarrow C_W = \frac{1}{W_{max}}$.
It is customary for NMF settings, to assume the two factor matrices *independent a priori*:

$$P(\mathbf{W}, \mathbf{H}) = P(\mathbf{W})P(\mathbf{H})$$

this states that without knowing $\mathbf{X}$ the knowledge of the matrix $\mathbf{H}$ does not improve our knowledge on the matrix $\mathbf{W}$.
Further a priori independence assumptions can be stated as:

$$P(\mathbf{W}) = \prod_i \prod_k P(W_{ik}) \tag{7.11}$$

$$P(\mathbf{H}) = \prod_k \prod_j P(H_{kj}) \tag{7.12}$$

Under these assumptions, NMF can be stated as MAP estimation maximizing the following quantity:

$$\begin{aligned} \ln \quad & ((P(\mathbf{X}|\mathbf{W},\mathbf{H})P(\mathbf{W},\mathbf{H})) \\ = \quad & \ln(P(\mathbf{X}|\mathbf{W},\mathbf{H})) + \ln(P(\mathbf{W},\mathbf{H})) \\ = \quad & \ln(P(\mathbf{X}|\mathbf{W},\mathbf{H})) + \ln(P(\mathbf{W})) + \ln(P(\mathbf{H})) \\ = \quad & \sum_i \sum_j \ln P(X_{ij}|\mathbf{W},\mathbf{H}) + \sum_i \sum_k \ln P(W_{ik}) + \sum_k \sum_j \ln P(H_{kj}) \end{aligned}$$

Note that the last independence assumptions are not necessary in general. One can also model the correlations between factors or properties within a factor (see e.g. [SL08]).

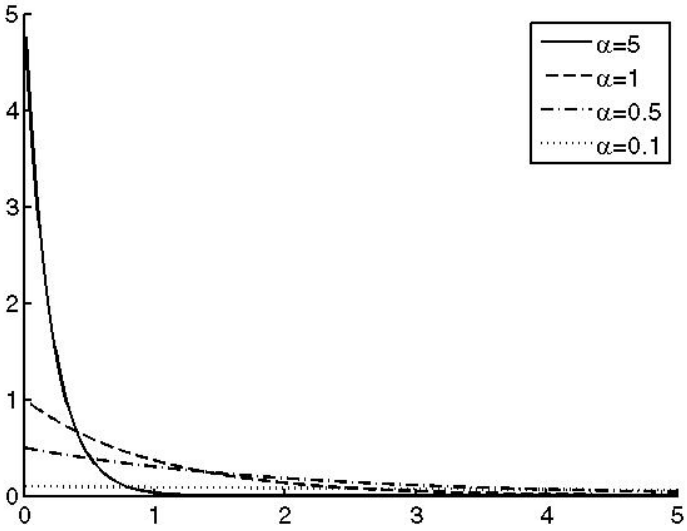

Figure 7.1: Examples for the exponential distribution function $p(t) = \alpha \exp(-\alpha t)$. The parameter $\alpha > 0$ determines the slope of the function. Large α leads to increased preference of small values.

Non-negative sparse coding

Non-negative sparse coding [Hoy02] as discussed in chapter 3 can be derived in a MAP framework under the following assumptions:

- $P(\mathbf{X}|\mathbf{W},\mathbf{H}) = \prod_i \prod_j \mathcal{N}(X_{ij}; [\mathbf{WH}]_{ij}, \sigma_r^2)$
- $P(\mathbf{W},\mathbf{H}) = P(\mathbf{W})P(\mathbf{H})$
- $P(\mathbf{W}) = \prod_i \prod_k \alpha \exp(-\alpha W_{ik}), \quad \alpha > 0$
- $P(H_{kj}) = \begin{cases} \text{const.}, & 0 \leq H_{kj} \leq H_{max} \\ 0, & \text{else.} \end{cases}$

MAP estimation means the maximization of the following quantity w.r.t $\mathbf{W}$ and $\mathbf{H}$

$$\ln(P(\mathbf{X}|\mathbf{W},\mathbf{H})) + \ln(P(\mathbf{W},\mathbf{H})) =$$
$$-\frac{1}{2\sigma_r^2} \sum_i \sum_j (X_{ij} - [\mathbf{WH}]_{ij})^2 - \alpha \sum_{ik} W_{ik} + \text{constant terms}$$

Multiplication by $-2\sigma_r^2$ and setting $\lambda := 2\sigma_r^2 \alpha \geq 0$ shows that this is equivalent to the minimization of

$$\sum_i \sum_j (X_{ij} - [\mathbf{WH}]_{ij})^2 + \lambda \sum_{ik} W_{ik} \tag{7.13}$$

which is essentially the cost function used in [Hoy02] for non-negative sparse coding. (This interpretation can also be found in [SL08]).
The idea of sparse coding is to encourage small values in the weight matrix $\mathbf{W}$. The larger the parameter α is set, the more are small values preferred over large ones since the exponential function is steeper decreasing (see figure 7.1). Thus, sparse representations can be are favored by introducing an exponential prior distribution on the weight matrix $\mathbf{W}$. Note that other choices are possible to enforce sparsity, e.g. a rectified Gaussian with mean parameter $\mu \leq 0$.

7.2 Bayesian Nonnegative Matrix Factorization

The incorporation of Bayesian techniques to NMF is a natural step. Since NMF has in general no unique solution (see the discussion in chapter 3) it is necessary to introduce some additional constraints reflecting our prior knowledge on the system. The Bayesian formalism offers a possibility to do so.
It is not accidental that the Richardson-Lucy algorithm which was first presented in 1972 in a paper titled *Bayesian-Based Iterative Method of Image Restoration* [Ric72] is one of the forefathers of modern NMF algorithms [CZPA09].
As discussed in the previous section, it is well known in the NMF-literature, that the quadratic cost function (2.4) can be derived by a maximum likelihood approach assuming additive Gaussian noise, while the KL-divergence for NMF and the Itakura-Saito divergence can also be placed in a probabilistic framework by maximum likelihood estimation assuming Poisson noise or multiplicative Gamma noise respectively (see e.g. [CZA06], [SL08], [FC09]).
Several papers explicitly suggest Bayesian techniques to incorporate prior knowledge on the factor matrices in NMF.
Moussaoui et al. [MBCMD04] present a Bayesian method for positive source separation and assign independent Gamma priors for sources and mixtures.
Gaussian process priors for NMF are proposed by Schmidt et al. [SL08]. The matrices $\mathbf{W}$ and $\mathbf{H}$ are assumed to be independently determined by a Gaussian process connected by a link function.
A slow variation in temporal structures is a desired property for the separation of audio signals and has been incorporated e.g. by Markov-chain inverse Gamma prior distributions. For example, Virtanen et al. [VCG08] propose Bayesian extensions for audio signal modeling and utilize a Poisson likelihood and Gamma chain priors to model the spectral smoothness in natural sounds. Further, Bertin et al. [BBV09] enforce spectral harmonicity by imposing that the basis components are expressed as linear combinations of fixed narrow-band harmonic spectra and temporal continuity of the weights.
In the above examples, the Bayesian approach means to use not only the likelihood but to incorporate prior knowledge on the sources and weights and perform MAP estimation. In MAP techniques, however, the actual value of the posterior distribution is not exploited since the location of the maximum is the only quantity of interest. In the introduction to Bayesian methods in chapter 6, we have seen that Bayesian techniques can provide much more than just constrained optimization. So called *full* Bayesian approaches which go beyond MAP estimation and utilize the Bayesian framework e.g. for model selection to NMF have also been studied recently: Schmidt et. al. [SWH09] demonstrate sampling schemes a Gaussian likelihood and exponential priors to evaluate the posterior distribution of parameters. Cemgil [Cem09] discusses a variational Bayes approach (as introduced in section 6.2) and sampling methods for the Poisson likelihood. Further, Févotte et al. [FC09] sketch the use of MCMC and variational Bayes methods to perform Bayesian inference on NMF problems using the Euclidean, KL-divergence and Itakura-Saito divergence. Tan and Févotte [TF09] discuss a technique called *automatic relevance determination* which exploit Bayesian inference to automatically determine the optimal number of components, given some suitable assumptions, in a KL-NMF setting.

7.2.1 Bayesian NMF

Bayesian non-negative matrix factorization (Schmidt et al. [SWH09]) assumes a Gaussian likelihood, independent exponential priors on $\mathbf{W}$ and $\mathbf{H}$ with scales α_{ik}, β_{ik} and an inverse gamma with shape k and scale θ prior distribution for the noise variance :

- $P(\mathbf{X}|\mathbf{\Theta}) = \prod_i \prod_j \mathcal{N}(X_{ij}; [\mathbf{WH}]_{ij}, \sigma_r^2)$
- $P(\mathbf{W}) = \prod_i \prod_k \alpha_{ik} \exp(-\alpha_{ik} W_{ik})$
- $P(\mathbf{H}) = \prod_k \prod_j \beta_{kj} \exp(-\beta_{kj} H_{kj})$

- $P(\sigma_r^2) = \frac{\theta^k}{\Gamma(k)}(\sigma_r^2)^{k-1}\exp(-\frac{\theta}{\sigma_r^2})$

According to Bayes' rule, the posterior distribution of all parameters in the model is given by

$$P(\mathbf{W}, \mathbf{H}, \sigma_\mathbf{r}|\mathbf{X}) \propto P(\mathbf{X}|\mathbf{W}, \mathbf{H}, \sigma_\mathbf{r})P(\mathbf{W})P(\mathbf{H})P(\sigma_\mathbf{r}) \tag{7.14}$$

The joint posterior density is approximated in a sampling scheme by iteratively sampling one parameter while keeping all others fixed. Expressions are derived for the conditional posterior densities of the model parameters $P(W_{ik}|\mathbf{X}, W_{\neg ik}, \mathbf{H}, \sigma_r^2)$, $P(H_{kj}|\mathbf{X}, \mathbf{W}, H_{\neg kj}, \sigma_r^2)$ and $P(\sigma_r^2|\mathbf{X}, \mathbf{W}, \mathbf{H})$, where the index $\neg ik$ depicts all entries of a matrix except entry ik.
A Gibbs sampler [GG87] is used to maximize the posterior density of the parameters by iteratively drawing samples from these conditional posterior distributions which converge towards the joint posterior distribution. Fortunately, there are closed forms of the densities to be drawn from and hence no samples need to be stored and the normalization constant can be computed. The authors demonstrate that the procedure is able to determine the correct number of components in a toy example and in a chemical shift imaging (CSI) dataset.
This method is related to a bilinear Bayesian approach [OSAMB99] for spectral decomposition which uses a modified commercial MCMC toolbox, and the Bayesian non-negative source separation method of Moussaoui et al. [MBMDC06] which incorporates a hybrid Gibbs-Metropolis-Hastings procedure. The direct Gibbs sampling procedure [SWH09] is faster than its precursors. Moreover, it addresses the problem of model order selection for NMF.

7.2.2 Automatic Relevance Determination

Tan and Févotte [TF09] discuss automatic relevance determination for KL-NMF for model order selection which does not need to evaluate the evidence by formulating a MAP criterion:

$$C_{MAP}(\mathbf{W}, \mathbf{H}, \beta) = -\ln P(\mathbf{W}, \mathbf{H}, \beta|\mathbf{X}) \tag{7.15}$$

$$= -\ln P(\mathbf{X}|\mathbf{W}, \mathbf{H}) - \ln P(\mathbf{W}|\beta) - \ln P(\mathbf{H}|\beta) - \ln P(\beta) \tag{7.16}$$

using KL-divergence log likelihood and independent half-normal priors on each column of $\mathbf{W}$ and row of $\mathbf{H}$ with precision parameter β_k

$$P(W_{ik}) = \sqrt{\frac{2}{\beta_k\pi}}\exp(-\frac{1}{2}\beta_k W_{ik}^2), \qquad W_{ik} \geq 0 \tag{7.17}$$

$$P(H_{kj}) = \sqrt{\frac{2}{\beta_k\pi}}\exp(-\frac{1}{2}\beta_k H_{kj}^2), \qquad H_{kj} \geq 0 \tag{7.18}$$

The precision parameters β_k are provided with a Gamma prior

$$P(\beta_k|a_k, b_k) = \frac{b_k^{a_k}}{\Gamma(a_k)}\beta_k^{a_k-1}\exp(-\beta_k b_k), \qquad \beta_k \geq 0 \tag{7.19}$$

with fixed hyperparameters a and b.
A multiplicative algorithm optimizes C_{MAP} in eq. (7.15) by iteratively updating $\mathbf{H}$, $\mathbf{W}$ and β.
The data automatically determines the optimal values of the hyperparameters β. The algorithm is initialized with a relatively large value K of components and successively drives unnecessary components to extinction. This property results from Bayesian inference: a subset of the precision parameters will be driven to an upper bound which corresponds to a sharp peak at zero for the priors on $\mathbf{W}_{*k}$ and row $\mathbf{H}_{k*}$ and leads to an effective extinction of column $\mathbf{W}_{*k}$ and row $\mathbf{H}_{k*}$. The effective number of components is determined by the number parameters β_k which are not driven to an upper bound during the iterations.
The effect of automatic relevance determination in Bayesian inference was earlier described by MacKay [Mac95] for supervised neural networks.

7.2.3 Bayesian Inference for Nonnegative Matrix factorization models

The paper of Cemgil [Cem09] nicely explains the statistical framework of NMF for the KL-divergence. It shows that adding Bayesian priors is equivalent to imposing regularization constraints to the NMF cost function and further demonstrates how the Bayesian formalism allows inference of further properties such as the best number of components.

The statistical perspective of KL-NMF

Suppose there are two sets of parameters $\mathbf{\Theta}^W$ and $\mathbf{\Theta}^H$ which parameterize some probability distributions $p(\mathbf{W}|\mathbf{\Theta}^W)$ and $p(\mathbf{H}|\mathbf{\Theta}^H)$.
First, the *latent sources* S_{ikj} are defined for $i = 1, \ldots, N$ $j = 1, \ldots, M$ $k = 1, \ldots, K$. Each variable S_{ikj} can be drawn from a Poisson distribution with parameter $W_{ik}H_{kj}$
Second, the observed data is given by

$$X_{ij} = \sum_k S_{ikj} \tag{7.20}$$

Hierarchical model

This leads to the following hierarchical model:

1. For $i = 1, \ldots, N$, $j = 1, \ldots, M$, $k = 1, \ldots, K$:
   Draw variables W_{ik} and H_{kj} from some parameterized prior distributions

$$W_{ik} \sim p(W_{ik}|\mathbf{\Theta}^W) \qquad \text{and} \qquad H_{kj} \sim p(H_{kj}|\mathbf{\Theta}^H) \tag{7.21}$$

2. For $i = 1, \ldots, N$, $j = 1, \ldots, M$, $k = 1, \ldots, K$:
   Draw variable S_{ikj} from a Poisson distribution

$$S_{ikj} \sim \exp\left(S_{ikj} \ln(W_{ik}H_{kj}) - W_{ik}H_{kj} - \ln\Gamma(S_{ikj} + 1)\right) \tag{7.22}$$

3. For $i = 1, \ldots, N$, $j = 1, \ldots, M$
   Compute

$$X_{ij} = \sum_k S_{ikj} \tag{7.23}$$

Maximum Likelihood

The log likelihood of the observed data $\mathbf{X}$ can be written as

$$LL_{\mathbf{X}}(\mathbf{W}, \mathbf{H}) =: \ln P(\mathbf{X}|\mathbf{W}, \mathbf{H}) = \ln \int_{\mathbf{S}} p(\mathbf{X}|\mathbf{S}) p(\mathbf{S}|\mathbf{W}, \mathbf{H}) d\mathbf{S} \tag{7.24}$$

where all possible values of the tensor $\mathbf{S}$ are integrated over or *marginalized out*. The logarithm can be brought inside the integral by introducing an instrumental distribution $q(\mathbf{S})$ where $\int_{\mathbf{S}} q(\mathbf{S}) d\mathbf{S} = 1$ and applying Jensen's inequality [Jen06]

$$LL_{\mathbf{X}}(\mathbf{W}, \mathbf{H}) \geq \int_{\mathbf{S}} q(\mathbf{S}) \ln \frac{p(\mathbf{X}|\mathbf{S}) p(\mathbf{S}|\mathbf{W}, \mathbf{H})}{q(\mathbf{S})} d\mathbf{S} \tag{7.25}$$

An *EM*-algorithm [DLR77] is applied which iteratively updates the latent variables $\mathbf{S}$ and parameters $\mathbf{W}$,$\mathbf{H}$ via

$$\text{E-step} \quad q(\mathbf{S}) = P(\mathbf{S}|\mathbf{X}, \mathbf{W}, \mathbf{H}) \tag{7.26}$$

$$\text{M-step} \quad (\mathbf{W}, \mathbf{H}) = \text{argmax}_{(\mathbf{W},\mathbf{H})} \left\langle \ln P(\mathbf{S}, \mathbf{X}|\mathbf{W}, \mathbf{H}) \right\rangle_{q(\mathbf{S})} \tag{7.27}$$

The resulting equations turn out to be identical to the multiplicative NMF update rules for KL-divergence [LS01], given in eq. 2.7 in chapter 2.

Full Bayesian Inference

Given the hyperparameters $\mathbf{\Theta} = \left\{\mathbf{\Theta}^W, \mathbf{\Theta}^H\right\}$ the evidence can be obtained by integrating over all hidden variables and parameters

$$\begin{aligned} P(\mathbf{X}|\mathbf{\Theta}) &= \int\int\int P(\mathbf{X},\mathbf{S},\mathbf{W},\mathbf{H}|\mathbf{\Theta})d\mathbf{W}d\mathbf{H}d\mathbf{S} && (7.28)\\ &= \int P(\mathbf{X}|\mathbf{S})P(\mathbf{S}|\mathbf{W},\mathbf{H})P(\mathbf{W},\mathbf{H}|\mathbf{\Theta})d\mathbf{W}d\mathbf{H}d\mathbf{S} && (7.29)\\ & && (7.30)\end{aligned}$$

A variational Bayes algorithm [GB00b],[Att00] as described in section 6.2 is derived which iteratively maximizes a lower bound $\mathcal{B}_{VB}$ given by

$$\mathcal{B}_{VB} = \int\int\int q \ln \frac{P(\mathbf{X},\mathbf{S},\mathbf{W},\mathbf{H}|\mathbf{\Theta})}{q} d\mathbf{W}d\mathbf{H}d\mathbf{S} \leq \ln P(\mathbf{X}|\mathbf{\Theta}) \tag{7.31}$$

by the updates

$$\begin{aligned} q(\mathbf{S}) &\propto \exp\left(\langle \ln P(\mathbf{X},\mathbf{S},\mathbf{W},\mathbf{H}|\mathbf{\Theta})\rangle_{q(\mathbf{W})q(\mathbf{H})}\right) && (7.32)\\ q(\mathbf{W}) &\propto \exp\left(\langle \ln P(\mathbf{X},\mathbf{S},\mathbf{W},\mathbf{H}|\mathbf{\Theta})\rangle_{q(\mathbf{S})q(\mathbf{H})}\right) && (7.33)\\ q(\mathbf{H}) &\propto \exp\left(\langle \ln P(\mathbf{X},\mathbf{S},\mathbf{W},\mathbf{H}|\mathbf{\Theta})\rangle_{q(\mathbf{S})q(\mathbf{W})}\right) && (7.34)\end{aligned}$$

The instrumental distribution q in eq. (7.31) is assumed to have a completely factorized form

$$q = q(\mathbf{S},\mathbf{W},\mathbf{H}) = \prod_i\prod_k\prod_j q(S_{ikj}) \prod_i\prod_k q(W_{ik}) \prod_k\prod_j H_{kj} \tag{7.35}$$

The necessary expectations are functions of the sufficient statistics of the variational distributions q and are derived for independent Gamma priors on W_{ik} and H_{kj} which are conjugate to the Poisson likelihood.
The paper further discusses a MCMC sampling algorithm which is omitted here.
Excellent results for model order selection according to the log evidence bound $\mathcal{B}_{VB}$ and the sampling algorithm on toy data and face images are presented.
This paper was discussed at length here since it remarkably summarizes the Bayesian approaches to NMF, and, as a by-product shows that the original Lee-Seung algorithm for KL-NMF is an EM algorithm and the variational Bayes algorithm a generalization thereof.
Févotte and Cemgil [FC09] present an EM algorithm for Euclidean NMF and mention that the update equations differ from the usual multiplicative Lee Seung updates [LS01].
In paragraph 8.2, a counterpart to the variational Bayes algorithm discussed here for Euclidean NMF using rectified Gaussian priors will be presented, whose maximum likelihood version turns out to be identical to the Lee-Seung algorithm.

Chapter 8

Bayesian extensions to NMF

Although non-negative matrix factorization has become a popular analysis tool for non-negative data sets (see the literature survey in chapter 2), there are still some open issues remaining partly unsolved. Two important problems concern the uniqueness (discussed in chapter 3) and the determination of the optimal number of underlying components.
The emergence of Bayesian techniques in combination with NMF (as reviewed in chapter 7.2) in the recent past shows promising results towards the solution of the mentioned problems.
Usually, Bayesian methods are used to incorporate prior knowledge into the NMF problem. The additional constraints expressed by prior distributions on the parameter matrices then induce some kind of uniqueness to the NMF solutions. In section 8.1 we will see what happens if we use the Bayesian formalism to express the absence of prior knowledge. A Bayesian optimality criterion for NMF solutions in is derived using a Laplace approximation. In the special case of a Gaussian likelihood a simple criterion arises from the general procedure.
A Gaussian likelihood counterpart to the variational Bayes NMF algorithm by Cemgil [Cem09] which was discussed in paragraph 7.2 is given in section 8.2.2 and its ability to determine the optimal number of components is investigated in simulations.
The Bayesian approach to NMF treats the factor matrices $\mathbf{W}$ and $\mathbf{H}$ as parameters, while the reconstruction error between the data $\mathbf{X}$ and its approximation $\mathbf{WH}$ is expressed by the logarithm of a likelihood function $P(\mathbf{X}|\mathbf{W},\mathbf{H})$. The likelihood function can contain additional parameters, such as the standard deviation σ_r in the Gaussian likelihood case. Further, prior distributions $P(\mathbf{W})$ and $P(\mathbf{H})$ have to be assigned which can also have extra parameters called hyperparameters. The Bayesian model for NMF is then specified by the likelihood function and the form of the priors plus the factorization rank K which is the number of columns/rows in $\mathbf{W}/\mathbf{H}$.
Once all assumptions have been stated properly, inference follows automatically by the famous Bayes' formula:

$$P(\mathbf{W},\mathbf{H},\mathbf{\Theta}|\mathbf{X},K)P(\mathbf{X}|K) = P(\mathbf{X}|\mathbf{W},\mathbf{H},\mathbf{\Theta})P(\mathbf{W},\mathbf{H},\mathbf{\Theta}|K) \tag{8.1}$$

where $\mathbf{\Theta}$ denotes eventually additional parameters.
In section 8.2.2 we will interpret different models by different factorization ranks K and see how Bayes can help to determine the optimal number of sources in a given data set under certain prior assumptions.

8.1 The Bayesian approach to Uniqueness

We discussed the uniqueness ambiguity of an exact NMF problem $\mathbf{X} = \mathbf{WH}$ in detail in chapter 3 from a geometrical point of view. Here, we build up on this and formulate the problem in a Bayesian

context by stating all assumptions at the beginning. Each observation is stored in a row of the data matrix $\mathbf{X}$. These M-dimensional data vectors $\mathbf{X}_{1*}, \ldots, \mathbf{X}_{N*} \geq 0$ lie in the non-negative orthant. We assume that each vector $\mathbf{X}_{i*}$ can be represented as a linear combination

$$\mathbf{X}_{i*} = \sum_{k=1}^{K} W_{ik}\mathbf{H}_{k*} + \mathbf{E}_{i*} \tag{8.2}$$

where $\mathbf{H}_{1*}, \ldots, \mathbf{H}_{K*} \geq 0$ are the hidden basis vectors and $W_{i1}, \ldots, W_{iK}$ are the weights which are necessary to describe vector $\mathbf{X}_{i*}$ in basis $\mathbf{H}$. We further assume the entries of the noise $\mathbf{E}_{i*}$ to be independent identical distributed according to a Gaussian with zero mean and standard deviation σ_r $\mathcal{N}(0, \sigma_r)$.
If the number of basis components K is smaller than the original data dimension M, the basis vectors span a K-dimensional subset of the M-dimensional non-negative orthant. $\mathbf{W} \geq 0$ means that the data $\mathbf{X}$ lies in the positive span of the vectors given by the rows of $\mathbf{H}$, or that the basis vectors enclose the data in a non-negative fashion. We assume the noise level σ_r to be small such that the association of basis vectors enclosing the data still holds approximately, and may be slightly violated by single outer points.
Subtracting the noise from each observation,

$$\mathbf{X}_{i*} - \mathbf{E}_{i*} = \sum_{k=1}^{K} W_{ik}\mathbf{H}_{k*} \tag{8.3}$$

the non-negative superposition model holds again. Note that the assumption of additive i.i.d. noise E_{ij} is only an approximation, since the left hand side of eq. (8.3) could be negative in principle if some $X_{ij} - E_{ij} < 0$. Since the right hand side of eq. (8.3) is non-negative by definition and the observed data is also non-negative, the noise is not independent from the data and small X_{ij} are related to small noise E_{ij} in practice.
Keeping the noise parameter σ_r small in mind, this can be neglected here.
The ambiguity of NMF is an intrinsic problem of the matrix multiplication since any invertible matrix $\mathbf{S}$ induces an alternative solution if $\mathbf{WS}^{-1} \geq 0$ and $\mathbf{SH} \geq 0$ since

$$\mathbf{X} = \mathbf{WH} = \mathbf{WS}^{-1}\mathbf{SH} \tag{8.4}$$

(see sections 3.1.2 and 3.2).

8.1.1 The most probable matrix H given X

Given a factorization rank K and a non-negative data set $\mathbf{X}$, we are interested in the most probable non-negative matrix $\mathbf{H}$ enclosing $\mathbf{X}$ via non-negative weights $\mathbf{W}$ in absence of any additional prior information. This requires the computation of the posterior distribution of $\mathbf{H}$, which can be written according to Bayes' rule as

$$P(\mathbf{H}|\mathbf{X}) = \frac{P(\mathbf{X}|\mathbf{H})P(\mathbf{H})}{P(\mathbf{X})} \tag{8.5}$$

We can formalize the absence of further knowledge on the values of $\mathbf{H}$ (except their non-negativity) by assuming a flat prior

$$P(H_{kj}) = \begin{cases} C_{H_k}, & \text{if } 0 \leq H_{kj} \leq H_{k,max} \\ 0 & \text{else.} \end{cases} \tag{8.6}$$

where C_{H_k} and $H_{k,max}$ are suitable non-negative constants such that the prior can be normalized $\int P(H_{kj})dH_{kj} = 1$.

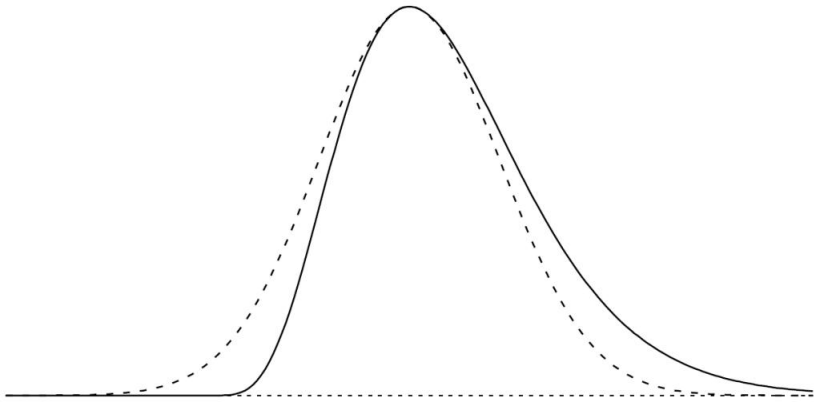

Figure 8.1: Laplace's method approximates a complicated function (solid) by a Gaussian around its maximum (dotted) to estimate its integral (taken from [Mac03])

Assuming independent flat priors of the form (8.6) for every H_{kj}, according to eq. (8.5) the posterior $P(\mathbf{H}|\mathbf{X})$, is proportional to $P(\mathbf{X}|\mathbf{H})$, which can be computed via

$$P(\mathbf{X}|\mathbf{H}) = \int P(\mathbf{X}|\mathbf{W},\mathbf{H})P(\mathbf{W})d\mathbf{W} \tag{8.7}$$

The last expression represents the likelihood integrated over all possible W_{ik} and involves NK dimensions.

Laplace's method

We can approximately solve the typically complicated integral in eq. (8.7) utilizing Laplace's method which is also known as saddle point approximation (see e.g. [Mac03]) as follows:
Let θ be a L-dimensional vector and $\tilde{P}(\theta)$ an un-normalized probability density. Expanding the logarithm of the integrand in

$$\int \tilde{P}(\theta)d\theta \tag{8.8}$$

by a Taylor series around its maximum θ^* yields

$$\ln \tilde{P}(\theta) \approx \ln \tilde{P}(\theta^*) - \frac{1}{2}(\theta-\theta^*)^T\mathbf{A}(\theta-\theta^*) + ... \tag{8.9}$$

where the first derivatives are zero due to the maximum condition and

$$A_{ij} = -\left.\frac{\partial^2}{\partial\theta_i\partial\theta_j}\ln\tilde{P}(\theta)\right|_{\theta=\theta^*}$$

is the negative $L \times L$ Hessian matrix collecting the second order partial derivatives w.r.t. the entries of vector θ.
Exponentiating both sides of eq. (8.9) yields

$$\tilde{P}(\theta) \approx \tilde{P}(\theta^*)\exp\left(-\frac{1}{2}(\theta-\theta^*)^T\mathbf{A}(\theta-\theta^*)\right) \tag{8.10}$$

and the original integral can be approximated by the well-known L-dimensional Gaussian

$$\int \tilde{P}(\theta)d\theta \approx \tilde{P}(\theta^*)\sqrt{\frac{(2\pi)^L}{\det(\mathbf{A})}} \tag{8.11}$$

8.1.2 Laplace's approximation for NMF

Utilizing Laplace's method, the integral (8.7) can be approximated by a Gaussian integral around the maximum of the integrand, yielding

$$P(\mathbf{X}|\mathbf{H}) \approx P(\mathbf{X}|\mathbf{W}^*, \mathbf{H})P(\mathbf{W}^*)(2\pi)^{\frac{NK}{2}} \det{}^{-\frac{1}{2}}(\mathbf{A}(\mathbf{W}^*)) \tag{8.12}$$

where

$$\mathbf{A}(\mathbf{W}^*) =: -\nabla_W \nabla_W \{\ln\left(P(\mathbf{X}|\mathbf{W}, \mathbf{H})P(\mathbf{W})\right)\}|_{\mathbf{W}=\mathbf{W}^*} \tag{8.13}$$

is a $NK \times NK$ matrix containing the negative second order derivatives w.r.t. all entries W_{ik} at the maximum.
Depending on the form of the likelihood $P(\mathbf{X}|\mathbf{W}, \mathbf{H})$ and prior $\mathbf{W}$, the second order Taylor approximation is more or less accurate. In a small surrounding of the maximum, it is always valid, but the integrand can in general have various shapes with more ore less dominating maxima (see Fig. 8.1).

8.1.3 The Bayesian optimality condition for Gaussian NMF

Here we evaluate the Laplace approximation (8.12) in case of a Gaussian likelihood and independent flat priors on $\mathbf{W}$ of the form.

$$P(W_{ik}) = \begin{cases} C_W, & \text{if } 0 \leq W_{ik} \leq W_{max} \\ 0 & \text{else.} \end{cases} \tag{8.14}$$

In case of the Gaussian likelihood for NMF

$$P(\mathbf{X}|\mathbf{W}, \mathbf{H}) = \prod_i \prod_j \frac{1}{\sqrt{2\pi}\sigma_r} e^{-\frac{1}{2}\left(\frac{X_{ij} - [\mathbf{WH}]_{ij}}{\sigma_r}\right)^2} \tag{8.15}$$

the first order derivatives of $\ln(P(\mathbf{X}|\mathbf{W}, \mathbf{H})P(\mathbf{W})$ w.r.t. the entries of $\mathbf{W}$ are given by

$$\frac{\partial}{\partial W_{ab}} \ln(P(\mathbf{X}|\mathbf{W}, \mathbf{H})P(\mathbf{W})) = \frac{1}{\sigma_r^2} \sum_j (X_{aj} - [WH]_{aj})H_{bj} \tag{8.16}$$

and the second order derivatives in eq. (8.13) are

$$\frac{\partial^2}{\partial W_{ab} \partial W_{cd}} \ln(P(\mathbf{X}|\mathbf{W}, \mathbf{H})P(\mathbf{W})) = -\frac{1}{\sigma_r^2} \sum_j H_{bj} H_{dj} \delta_{ac} \tag{8.17}$$

leading to the following negative Hessian matrix of size $NK \times NK$:

$$\mathbf{A} = \frac{1}{\sigma_r^2} \left(\begin{array}{ccc|c|ccc} [HH^T]_{11} & 0 & 0 & \cdots & [HH^T]_{1K} & 0 & 0 \\ 0 & \ddots & 0 & \cdots & 0 & \ddots & 0 \\ 0 & 0 & [HH^T]_{11} & \cdots & 0 & 0 & [HH^T]_{1K} \\ \hline [HH^T]_{12} & 0 & 0 & \cdots & [HH^T]_{2K} & 0 & 0 \\ 0 & \ddots & 0 & \cdots & 0 & \ddots & 0 \\ 0 & 0 & [HH^T]_{12} & \cdots & 0 & 0 & [HH^T]_{2K} \\ \hline & \vdots & & \ddots & & \vdots & \\ \hline [HH^T]_{1K} & 0 & 0 & \cdots & [HH^T]_{KK} & 0 & 0 \\ 0 & \ddots & 0 & \cdots & 0 & \ddots & 0 \\ 0 & 0 & [HH^T]_{1K} & \cdots & 0 & 0 & [HH^T]_{KK} \end{array} \right)$$

where we exploited the symmetry of the matrix $\mathbf{H}\mathbf{H}^T = (\mathbf{H}\mathbf{H}^T)^T$.
We can calculate the required $\det(\mathbf{A})$ using standard manipulations from linear algebra as follows. For simplicity, we denote the matrix $\tilde{\mathbf{A}} := \sigma_r^2\mathbf{A}$. $\tilde{\mathbf{A}}$ can be brought to upper triangular form $\tilde{\mathbf{A}}^{\triangleleft}$ by elementary row operations as shown in the sequel.

Computing the triangular form

We will sketch the operations to bring $\tilde{\mathbf{A}}$ to upper triangular form $\tilde{\mathbf{A}}^{\triangleleft}$. The first entry to be eliminated is $[HH^T]_{12}$ on pos. (N+1,1). Subtracting

$$\frac{[HH^T]_{12}}{[HH^T]_{11}} \times \text{ row 1}$$

from row $(N+1)$ yields a new diagonal entry

$$[HH^T]_{22} - \frac{([HH^T]_{12})^2}{[HH^T]_{11}} \tag{8.18}$$

on pos. (N+1,N+1) and zeros on positions (N+1,1),...,(N+1,N) of the new intermediate matrix. The second entry of interest is $[HH^T]_{12}$ on pos. (N+2,2). It can be eliminated by subtracting

$$\frac{[HH^T]_{12}}{[HH^T]_{11}} \times \text{ row 2}$$

from row $(N+2)$, ending up with zeros on positions (N+2,1),...,(N+2,N+1) and a diagonal entry $(N+2, N+2)$ given also by eq. (8.18). This scheme can be continued for the whole block $(N+1, 1) \dots (2N, N)$, and leads to N identical diagonal entries of the form (8.18).
In fact, the whole matrix $\tilde{\mathbf{A}}$ can be brought to an upper triangular form $\tilde{\mathbf{A}}^{\triangleleft}$ by exactly the same row manipulations which are necessary when transforming the simpler $K \times K$ matrix $\mathbf{H}\mathbf{H}^T$ to its triangular form $(\mathbf{H}\mathbf{H}^T)^{\triangleleft}$. Moreover, the resulting diagonal entries of $\tilde{\mathbf{A}}^{\triangleleft}$ are N respective copies of the diagonal entries of the corresponding triangular matrix $(\mathbf{H}\mathbf{H}^T)^{\triangleleft}$.
Let $\lambda_1, \dots, \lambda_K$ denote the eigenvalues of $\mathbf{H}\mathbf{H}^T$. Then then $\tilde{\mathbf{A}}$ has the eigenvalues $(\lambda_1)^N, \dots, (\lambda_K)^N$, and

$$\det(\tilde{\mathbf{A}}) = \prod_{k=1}^{K} (\lambda_k)^N = (\prod_{k=1}^{K} \lambda_k)^N = (\det(\mathbf{H}\mathbf{H}^T))^N \tag{8.19}$$

and

$$\det(\mathbf{A}) = \det(\sigma_r^{-2}\tilde{\mathbf{A}}) = \sigma_r^{-2NK}\det(\tilde{\mathbf{A}}) = \sigma_r^{-2NK}(\det(\mathbf{H}\mathbf{H}^T))^N \tag{8.20}$$

The matrix $\mathbf{A}$ is well-behaved if its smaller counterpart $\mathbf{H}\mathbf{H}^T$ does so. This is the case if the factorization rank is at least K.
Note further that in the current case (Gaussian likelihood and independent flat priors on $\mathbf{W}$) all terms in the Taylor expansion of order higher than two are zero and the expression given by eq. (8.12) becomes an exact equality.
Taking the logarithm of eq. (8.12) and plugging in the result above,

$$\begin{aligned}\ln P(\mathbf{X}|\mathbf{H}) = &-\frac{1}{2\sigma_r^2}\sum_i\sum_j (X_{ij} - [\mathbf{W}^*\mathbf{H}]_{ij})^2 - \frac{N}{2}\ln\det(\mathbf{H}\mathbf{H}^T) \\ &+ NK\ln C_W + \frac{NK}{2}\ln(2\pi) + NK\ln\sigma_r\end{aligned} \tag{8.21}$$

The maximum of the above quantity w.r.t. $\mathbf{H}$ is the MAP estimate w.r.t. $\mathbf{W}$ and $\mathbf{H}$ since the expansion is around the maximum value $\mathbf{W}^*$.
Neglecting constants w.r.t. $\mathbf{W}$ and $\mathbf{H}$, the optimal solutions of a given NMF problem in absence of any prior knowledge on the factor matrices except non-negativity are given by those $\mathbf{H} \geq 0, \mathbf{W} \geq 0$ for which

$$BOC := -\frac{1}{2\sigma^2}\sum_i\sum_j (X_{ij} - [\mathbf{W}\mathbf{H}]_{ij})^2 - \frac{N}{2}\ln\det(\mathbf{H}\mathbf{H}^T) \tag{8.22}$$

is maximal.
We call the just derived criterion the *Bayesian optimality condition for Gaussian NMF*.

Geometrical interpretation

Geometrically, the determinant $\det(\mathbf{H}\mathbf{H}^T)$ describes the volume spanned by the vectors $\mathbf{H}_{1*}, \ldots, \mathbf{H}_{K*}$. the last term in eq. (8.22) can thus be interpreted as a penalty term constraining large volumes. Remarkably, the *BOC* describes the same solution as the one suggested by the *determinant criterion* criterion in chapter 3, where the geometrically motivated regularization term $\alpha\det(\mathbf{H}\mathbf{H}^T)$ was appended to the squared cost function and yielding a constrained minimization problem in eq. (3.8). The geometrical motivation stipulated that the basis vectors should only span regions of the data space if the observed data necessitates it. Due to the enclosing by non-negative weights $\mathbf{W}$, this implies that the volume spanned by the basis vectors $\mathbf{H}_{1*}, \ldots, \mathbf{H}_{K*}$ is minimal (under a normalization constraint to exclude trivial solutions).
Concluding, we show that the *detNMF* algorithm introduced in paragraph (3.2.4) in each step decreases an upper bound to the negative of the *BOC* given by eq. (8.22).

Relation between the *BOC* and the *detNMF* criterion

The function $f(t) = \ln(t)$ is concave and hence is always bounded from above by a linear Taylor approximation:

$$\ln(t) \leq \ln t_0 + \frac{t - t_0}{t_0}, \text{ for any } t_0 > 0 \tag{8.23}$$

Setting $a = \det^{-1}(\mathbf{H}_0\mathbf{H}_0^T)$ and $b = \ln\det(\mathbf{H}_0{\mathbf{H}_0}^T) - 1$ which are constants w.r.t. $\mathbf{H}$ for some fixed $\mathbf{H}_0$, we have

$$\ln\det(\mathbf{H}\mathbf{H}^T) \leq a\det(\mathbf{H}\mathbf{H}^T) + b \tag{8.24}$$

and thus

$$-\sigma_r^2 BOC = \sum_i \sum_j (X_{ij} - [\mathbf{WH}]_{ij})^2 + N\sigma_r^2 \ln\det(\mathbf{HH}^T) \tag{8.25}$$

$$\leq \sum_i \sum_j (X_{ij} - [\mathbf{WH}]_{ij})^2 + N\sigma_r^2 (a \det(\mathbf{HH}^T) + b) \tag{8.26}$$

Minimization of the latter expression implies maximization of eq. (8.22). Dropping the last constant w.r.t. $\mathbf{H}$, and setting $\alpha = N\sigma_r^2 \det^{-1}(\mathbf{H}_0\mathbf{H}_0^T)$ we see that this is equivalent to the minimization of

$$\sum_i \sum_j (X_{ij} - [\mathbf{WH}]_{ij})^2 + \alpha \det \mathbf{HH}^T \tag{8.27}$$

which is exactly the *detNMF* cost function 3.8 proposed in chapter 3.
The parameter α depends on the number of observations N, the noise parameter σ_r and the actual value $\mathbf{H}^0$.

8.2 Variational methods

After we introduced the basic concept of variational methods in paragraph (6.2) and Cemgil's application thereof to Kullback-Leibler NMF (7.2.3), this section presents an analogon for Euclidean NMF. Before we derive the Bayesian NMF algorithm, a maximum likelihood version is discussed which can be seen as an introductory exercise for the later generalization.

8.2.1 Maximum likelihood

Here we show that the multiplicative update rules for Euclidean NMF presented by Lee and Seung [LS01] can be interpreted as a variational maximum likelihood technique which optimizes a Jensen [Jen06] bound.
Starting with the squared Euclidean distance, we utilize the convexity of the quadratic function and derive an upper bound

$$\begin{aligned} E &= \frac{1}{2}\sum_i\sum_j (X_{ij} - [\mathbf{WH}]_{ij})^2 \qquad (8.28)\\ &= \frac{1}{2}\sum_i\sum_j \left(X_{ij}^2 - 2X_{ij}[\mathbf{WH}]_{ij} + \left(\sum_k W_{ik}H_{kj}\right)^2\right)\\ &= \frac{1}{2}\sum_i\sum_j \left(X_{ij}^2 - 2X_{ij}[\mathbf{WH}]_{ij} + \left(\sum_k q_{ikj}\frac{W_{ik}H_{kj}}{q_{ikj}}\right)^2\right)\\ &\leq \frac{1}{2}\sum_i\sum_j \left(X_{ij}^2 - 2X_{ij}[\mathbf{WH}]_{ij} + \sum_k q_{ikj}\left(\frac{W_{ik}H_{kj}}{q_{ikj}}\right)^2\right)\\ &= \frac{1}{2}\sum_i\sum_j \left(X_{ij}^2 - 2X_{ij}[\mathbf{WH}]_{ij} + \sum_k \frac{1}{q_{ikj}}\left(W_{ik}H_{kj}\right)^2\right) =: E_q \qquad (8.29)\end{aligned}$$

where q_{ikj} are auxiliary variables obeying

$$\sum_k q_{ikj} = 1 \text{ for all } i,j \text{ respectively.} \qquad (8.30)$$

The partial derivative of the plain cost function E in eq. (8.28) w.r.t one entry H_{kj}

$$\frac{\partial E}{\partial H_{kj}} = -\sum_i (X_{ij} - [\mathbf{WH}]_{ij})W_{ik} \qquad (8.31)$$

depends on the whole column $\mathbf{H}_{*j}$ and thus a usual gradient algorithm leads to inter-dependent parameters during one update.
Introducing the auxiliary variables q_{ikj} leads to a decoupling of the H_{kj}: The partial derivative of the bound E_q (8.29) w.r.t. H_{kj}

$$\frac{\partial E_q}{\partial H_{kj}} = -\sum_i (X_{ij} - \frac{W_{ik}H_{kj}}{q_{ikj}})W_{ik} \qquad (8.32)$$

does not depend on other entries of $\mathbf{H}$ except H_{kj}.

optimal q_{ikj}

The auxiliary variables q_{ikj} can be chosen such as to make the bound as tight as possible by setting

$$\frac{\partial}{\partial q_{ikj}} \left(E_q + \lambda_{ij} \left(\sum_k q_{ikj} - 1 \right) \right) = 0 \tag{8.33}$$

where the Lagrange multipliers λ_{ij} ensure the sum-to-one constraint (8.30). Solving (8.33) yields

$$\begin{aligned} 0 &= -\frac{(W_{ik}H_{kj})^2}{2q_{ikj}^2} + \lambda_{ij} \\ \Leftrightarrow q_{ikj} &= \frac{W_{ik}H_{kj}}{\sqrt{2\lambda_{ij}}} \end{aligned}$$

The Lagrange parameters λ_{ij} can be eliminated using the constraint (8.30)

$$\begin{aligned} 1 &= \sum_k q_{ikj} = \sum_k \frac{W_{ik}H_{kj}}{\sqrt{2\lambda_{ij}}} \\ \Leftrightarrow \sqrt{2\lambda_{ij}} &= \sum_k W_{ik}H_{kj} \end{aligned}$$

and finally

$$q_{ikj} = \frac{W_{ik}H_{kj}}{\sum_k W_{ik}H_{kj}} \tag{8.34}$$

There is an intuitive interpretation of the optimal auxiliary variables q_{ikj}: Since $X_{ij} \approx [\mathbf{WH}]_{ij}$ is the j'th dimension of variable i, from eq. (8.34) we see that q_{ikj} is the contribution of the k'th source to $[\mathbf{WH}]_{ij}$, normalized by the contributions of all K sources $[\mathbf{WH}]_{ij}$.

Hence, there is a clear relation to the classic EM- algorithm [DLR77] if we interpret the source contributions q_{ikj} as hidden variables and we alternate between updating the hidden variables q_{ikj} (E-step) and the parameters H_{kj} and W_{ik} (M-step).

This general interpretation of the EM- algorithm as lower bound maximization was mentioned in [NH98] and [Min98].

An iterative procedure which is guaranteed not to increase the bound E_q then alternates between the following steps:

$$\text{Solve for all } i,k,j \qquad 0 \stackrel{!}{=} \frac{\partial}{\partial q_{ikj}} \left(E_q + \lambda_{ij}(\sum_k q_{ikj} - 1) \right) \tag{8.35}$$

$$\text{Solve for all } i,k \qquad 0 \stackrel{!}{=} \frac{\partial E_q}{\partial W_{ik}} \tag{8.36}$$

$$\text{Solve for all } k,j \qquad 0 \stackrel{!}{=} \frac{\partial E_q}{\partial H_{kj}} \tag{8.37}$$

optimal H_{kj}

The optimal H_{kj} are gained by solving

$$\begin{aligned} 0 &= \frac{\partial E_q}{\partial H_{kj}} \\ &= -\sum_i X_{ij} W_{ik} + \sum_i H_{kj} \frac{W_{ik}^2}{q_{ikj}} \\ \Leftrightarrow H_{kj} &= \frac{\sum_i X_{ij} W_{ik}}{\sum_i \frac{W_{ik}^2}{q_{ikj}}} \end{aligned} \tag{8.38}$$

Plugging in the optimal q_{ikj} from eq. (8.34) yields

$$H_{kj} = \frac{\sum_i X_{ij} W_{ik}}{\sum_i \frac{W_{ik}^2 \sum_l W_{il} H_{lj}}{W_{ik} H_{kj}}} \tag{8.39}$$

$$= H_{kj} \frac{\sum_i X_{ij} W_{ik}}{\sum_i W_{ik} \sum_l W_{il} H_{lj}} \tag{8.40}$$

$$= H_{kj} \frac{[\mathbf{W}^T \mathbf{X}]_{kj}}{[\mathbf{W}^T \mathbf{W} \mathbf{H}]_{kj}} \tag{8.41}$$

optimal W_{ik}

In straight analogy, we can compute the optimal W_{ik};

$$0 = \frac{\partial E_q}{\partial W_{ik}} \tag{8.42}$$

$$= -\sum_j X_{ij} H_{kj} + \sum_j W_{ik} \frac{H_{kj}^2}{q_{ikj}} \tag{8.43}$$

$$\Leftrightarrow W_{ik} = \frac{\sum_j X_{ij} H_{kj}}{\sum_i \frac{H_{kj}^2}{q_{ikj}}} \tag{8.44}$$

$$= W_{ik} \frac{[\mathbf{X} \mathbf{H}^T]_{ik}}{[\mathbf{W} \mathbf{H} \mathbf{H}^T]_{ik}} \tag{8.45}$$

Equations (8.45) and (8.41) constitute the well-known Lee-Seung algorithm [LS99] for Euclidean NMF which we discussed already in the introductory NMF chapter 2, eq. (2.6).
We have just shown that the Lee-Seung algorithm for Euclidean NMF is generalized EM algorithm, where the updates are performed in the following order:

1. update all q_{ikj} via equation 8.34
2. update all H_{kj} via equation 8.38
3. update all q_{ikj} via equation 8.34
4. update all W_{ik} via equation 8.44

Steps (1) and (3) constitute a variational E-step, which computes an upper bound of E_q and minimizes it, while steps (2) and (4) are usual M-steps which update the parameters.

This example is discussed in detail as it nicely illustrates the generalized EM algorithm as bound optimization instead of the cost function itself.
Note that the non-negativity constraints on $\mathbf{W}$ and $\mathbf{H}$ need not be mentioned at any point and are conserved just by the multiplicative structure. The parameters need only be initialized by non-negative values.

8.2.2 Variational Bayes for Gaussian NMF

This section presents a variational Bayes algorithm for NMF under assumptions of a Gaussian likelihood and rectified Gaussian prior distributions. It can be interpreted as a variational Bayes extension to the Euclidean NMF algorithm by Lee and Seung [LS99] in its generalized EM interpretation given in the preceding paragraph (8.2.1). Here, probability distributions on the parameter matrices and their expectations are treated rather than point estimates in ML or MAP estimation techniques.
A similar couple of a maximum likelihood method together with its variational Bayesian extension was discussed for the Aspect Bernoulli model by Bingham and Kabán et al. ([KBH04], [BKF09], [KB08], see also paragraph (5.6.2)).
Variational methods have been extensively used for Bayesian learning tasks (see also [JGJS98], [Bis99], [Att00], [GB01], [HK07], [KB08] for examples).
To keep this section self-contained, we will give the whole derivation here, although some issues have already been discussed in preceding sections (e.g. in paragraphs 6.2, 7.2).
The main goal is to find a tractable approximation of the log evidence term

$$\ln P(\mathbf{X}) = \ln \int P(\mathbf{X}|\mathbf{W},\mathbf{H})P(\mathbf{W},\mathbf{H})d\mathbf{W}d\mathbf{H} \tag{8.46}$$

which requires $K(N+M)$ integrations over all parameters H_{kj} and W_{ik}.
The first step is to bring the logarithm into the integral. This can be done by introduction of a variational probability density $Q(\mathbf{W},\mathbf{H})$ such that

$$\int Q(\mathbf{W},\mathbf{H})d\mathbf{W}d\mathbf{H} = 1 \tag{8.47}$$

and application of Jensen's inequality [Jen06]

$$\begin{aligned} \ln P(\mathbf{X}) &\geq \int Q(\mathbf{W},\mathbf{H}) \ln \frac{P(\mathbf{X}|\mathbf{W},\mathbf{H})P(\mathbf{W},\mathbf{H})}{Q(\mathbf{W},\mathbf{H})} d\mathbf{W}d\mathbf{H} \\ &= \langle \ln P(\mathbf{X}|\mathbf{W},\mathbf{H})\rangle_{Q(\mathbf{W},\mathbf{H})} + \left\langle \ln \frac{P(\mathbf{W},\mathbf{H})}{Q(\mathbf{W},\mathbf{H})} \right\rangle_{Q(\mathbf{W},\mathbf{H})} := \mathcal{B}_Q \end{aligned} \tag{8.48}$$

where $\langle\cdot\rangle_Q$ denotes the expectation w.r.t. Q.
It can be shown that the lower bound $\mathcal{B}_Q$ equals the log evidence $\ln P(\mathbf{X})$ if Q is the joint posterior distribution of the parameters, $Q(\mathbf{W},\mathbf{H}) = P(\mathbf{W},\mathbf{H}|\mathbf{X})$ (see section 6.2, [Gha04])
This posterior is intractable in general, but sampling procedures can be applied to approximate it (see [SWH09]). Here, we follow a variational approach which is a variant of the *Variational Bayesian EM Algorithm* [Att00], [GB01] which was introduced in paragraph 6.2.2.
The following simplifying assumptions are made:

$$Q(\mathbf{W},\mathbf{H}) = \prod_i \prod_k Q(W_{ik}) \prod_k \prod_j Q(H_{kj}) \tag{8.49}$$

$$P(\mathbf{W},\mathbf{H}) = \prod_i \prod_k P(W_{ik}) \prod_k \prod_j P(H_{kj}) \tag{8.50}$$

The first assumption (8.49) implies that equality in (8.48) cannot be reached in general. Instead an approximation of the actual posterior distribution which is completely factorized will be computed. The second assumption (8.50) expresses that the parameters are independent a priori. This limitation is not necessary in general, but reasonable if we do not have explicit prior knowledge. A priori independence means here that knowledge of one parameter H_{kj} does not contain additional information about any other parameter's value before the data arrives. In the NMF model, all dependencies are modeled by the matrix product $\mathbf{X} \approx \mathbf{WH}$, and the a priori independence assumption is suggestive and will simplify the computations.
We further assume the Gaussian likelihood for NMF

$$P(\mathbf{X}|\mathbf{\Theta}) = \prod_i \prod_j \mathcal{N}(X_{ij}; [\mathbf{WH}]_{ij}, \sigma_r^2) \tag{8.51}$$

Putting everything together, the bound $\mathcal{B}_Q$ from eq. 8.48 takes the form

$$\begin{aligned}
\mathcal{B}_Q = \quad & - \quad \frac{NM}{2}\ln(2\pi) - NM\ln\sigma_r \\
& - \quad \frac{1}{2\sigma_r^2}\sum_i\sum_j\{X_{ij}^2 \\
& \qquad -2X_{ij}\sum_k \langle W_{ik}\rangle_{Q(W_{ik})}\langle H_{kj}\rangle_{Q(H_{kj})} \\
& \qquad + \left\langle([\mathbf{WH}]_{ij})^2\right\rangle_{Q(\mathbf{W}),Q(\mathbf{H})}\} \\
& + \quad \sum_i\sum_k\left\langle \ln\frac{P(W_{ik})}{Q(W_{ik})}\right\rangle_{Q(W_{ik})} + \sum_k\sum_j\left\langle \ln\frac{P(H_{kj})}{Q(H_{kj})}\right\rangle_{Q(H_{kj})}
\end{aligned} \tag{8.52}$$

Equation (8.52) contains coupled expectations of the form $\left\langle([\mathbf{WH}]_{ij})^2\right\rangle_{Q(\mathbf{W}),Q(\mathbf{H})}$. Application of Jensen's inequality once more by the same trick which was used in eq. (8.29) in the maximum likelihood version decouples these terms and leads to an additional lower bound

$$\begin{aligned}
\mathcal{B}_Q \geq \quad & - \quad \frac{NM}{2}\ln(2\pi) - NM\ln\sigma_r \\
& - \quad \frac{1}{2\sigma_r^2}\sum_i\sum_j\{X_{ij}^2 \\
& \qquad -2X_{ij}\sum_k \langle W_{ik}\rangle_{Q(W_{ik})}\langle H_{kj}\rangle_{Q(H_{kj})} \\
& \qquad + \sum_k \frac{1}{q_{ikj}}\left\langle(W_{ik})^2\right\rangle_{Q(W_{ik})}\left\langle(H_{kj})^2\right\rangle_{Q(H_{kj})}\} \\
& + \quad \sum_i\sum_k\left\langle \ln\frac{P(W_{ik})}{Q(W_{ik})}\right\rangle_{Q(W_{ik})} \\
& + \quad \sum_k\sum_j\left\langle \ln\frac{P(H_{kj})}{Q(H_{kj})}\right\rangle_{Q(H_{kj})} \\
=: \quad & \mathcal{B}_q
\end{aligned} \tag{8.53}$$

The price to pay for the decoupling comprises additional parameters q_{ikj} such that $\sum_k q_{ikj} = 1$ for every observation ij. $\mathcal{B}_q$ is always a lower bound to the log evidence and we gain the best approximation to $\ln P(\mathbf{X})$ if we maximize it w.r.t. all variational distributions $Q(W_{ik})$ and $Q(H_{kj})$ as well as the decoupling parameters q_{ikj}.

Note that a strict Bayesian approach would require assigning a prior distribution to the noise parameter σ_r as well and performing one more integration. Following the discussion in [Mac92a], we assume that the distribution $P(\sigma_r|\mathbf{X},\mathbf{W},\mathbf{H})$ is sharply peaked at the maximum. Thus, we can maximize $\mathcal{B}_q$ w.r.t. σ_r instead of performing another integral.
The optimal variational densities can be obtained by solving the following equations using variational calculus (see e.g. [Bis96] for a brief introduction to the necessary background)

$$0 \stackrel{!}{=} \frac{\partial}{\partial Q(H_{kj})}\left(\mathcal{B}_q - \lambda_{kj}^H\left(\int Q(H_{kj})dQ(H_{kj}) - 1\right)\right) \tag{8.54}$$

$$0 \stackrel{!}{=} \frac{\partial}{\partial Q(W_{ik})}\left(\mathcal{B}_q - \lambda_{ik}^W\left(\int Q(W_{ik})dQ(W_{ik}) - 1\right)\right) \tag{8.55}$$

where λ_{kj}^H and λ_{ik}^W are Lagrange parameters ensuring that $Q(H_{kj})$ and $Q(W_{ik})$ are densities. Further, the bound has to be maximized w.r.t. to the additional parameters by solving

$$0 \stackrel{!}{=} \frac{\partial}{\partial q_{ikj}}\left(\mathcal{B}_q - \lambda_{ikj}^q\left(\sum_k q_{ikj} - 1\right)\right) \tag{8.56}$$

with Lagrange multipliers λ_{ikj}^q.
Straightforward computation (see appendix C.1) leads to the following form for the optimal $Q(H_{kj})$:

$$Q(H_{kj}) \propto P(H_{kj}) \exp\left(\alpha_{kj}^H (H_{kj})^2 + \beta_{kj}^H H_{kj}\right) \tag{8.57}$$

where

$$\alpha_{kj}^H = -\frac{1}{2\sigma_r^2}\sum_i \frac{1}{q_{ikj}} \left\langle (W_{ik})^2 \right\rangle_{Q(W_{ik})} \tag{8.58}$$

and

$$\beta_{kj}^H = \frac{1}{\sigma_r^2}\sum_i X_{ij} \left\langle W_{ik} \right\rangle_{Q(W_{ik})} \tag{8.59}$$

The optimal $Q(W_{ik})$ can be computed in complete analogy:

$$Q(W_{ik}) \propto P(W_{ik}) \exp\left(\alpha_{ik}^W (W_{ik})^2 + \beta_{ik}^W W_{ik}\right) \tag{8.60}$$

where

$$\alpha_{ik}^W = -\frac{1}{2\sigma_r^2}\sum_j \frac{1}{q_{ikj}} \left\langle (H_{kj})^2 \right\rangle_{Q(H_{kj})} \tag{8.61}$$

and

$$\beta_{ik}^W = \frac{1}{\sigma_r^2}\sum_j X_{ij} \left\langle H_{kj} \right\rangle_{Q(H_{kj})} \tag{8.62}$$

Note that the distribution $Q(H_{kj})$ depends on the first and second order expectations $\langle W_{ik}\rangle_{Q(W_{ik})}$ and $\left\langle (W_{ik})^2 \right\rangle_{Q(W_{ik})}$ w.r.t. $Q(W_{ik})$ and vice versa.

The decoupling parameters q_{ikj} make the bound as tight as possible if they are set to

$$q_{ikj} = \frac{\sqrt{\langle (W_{ik})^2 \rangle_{Q(W_{ik})} \langle (H_{kj})^2 \rangle_{Q(H_{kj})}}}{\sum_k \sqrt{\langle (W_{ik})^2 \rangle_{Q(W_{ik})} \langle (H_{kj})^2 \rangle_{Q(H_{kj})}}} \tag{8.63}$$

Further, the optimal noise parameter σ_r is given by

$$\begin{aligned} \sigma_r^2 = & \frac{1}{NM} \sum_i \sum_j \Big\{ X_{ij}^2 \\ & - 2X_{ij} \sum_k \langle W_{ik} \rangle_{Q(W_{ik})} \langle H_{kj} \rangle_{Q(H_{kj})} \\ & + \sum_k \frac{1}{q_{ikj}} \langle (W_{ik})^2 \rangle_{Q(W_{ik})} \langle (H_{kj})^2 \rangle_{Q(H_{kj})} \Big\} \end{aligned} \tag{8.64}$$

(see appendix (C.1) for the derivations).
Before the expectations w.r.t. the distributions $Q(W_{ik})$, $Q(H_{kj})$ can be evaluated, the prior distributions $P(W_{ik})$, $P(H_{kj})$ need to be specified. Here we utilize the concept of conjugate priors as discussed in paragraph (6.1.3).

Conjugate priors

If we assume the prior on H_{kj} to be of the form

$$P(H_{kj}) \propto \exp\left((\alpha_{k0}^H (H_{kj})^2 + \beta_{k0}^H H_{kj} \right) \tag{8.65}$$

then the variational distribution will retain the same functional form

$$Q(H_{kj}) \propto \exp\left((\alpha_{k0}^H + \alpha_{kj}^H)(H_{kj})^2 + (\beta_{k0}^H + \beta_{kj}^H) H_{kj} \right) \tag{8.66}$$

This property is known as *conjugacy* in Bayesian literature [RS61]. Of course, we could chose other forms, but conjugacy renders the computations easier.

The rectified Gaussian distribution

A prior of the form (8.65) which is nonzero on the non-negative reals only is given by the rectified Gaussian distribution. It evolves from a usual Gaussian by truncating all negative entries an renormalizing the integral to one.

$$\mathcal{N}^+(\theta|\mu,\sigma) =: \begin{cases} \frac{1}{Z_{\mathcal{N}^+}} e^{-\frac{1}{2}\left(\frac{\theta-\mu}{\sigma}\right)^2}, & 0 \le \theta < \infty \\ 0 & \text{otherwise} \end{cases} \tag{8.67}$$

The normalization constant is given by

$$Z_{\mathcal{N}^+} = \frac{1}{2}\sigma\sqrt{2\pi}\ \mathrm{erfc}\left(-\frac{\mu}{\sigma\sqrt{2}} \right) \tag{8.68}$$

where

$$\mathrm{erfc}(z) =: \frac{2}{\sqrt{\pi}} \int_z^\infty \exp(-t^2) dt \tag{8.69}$$

is the complementary error function.

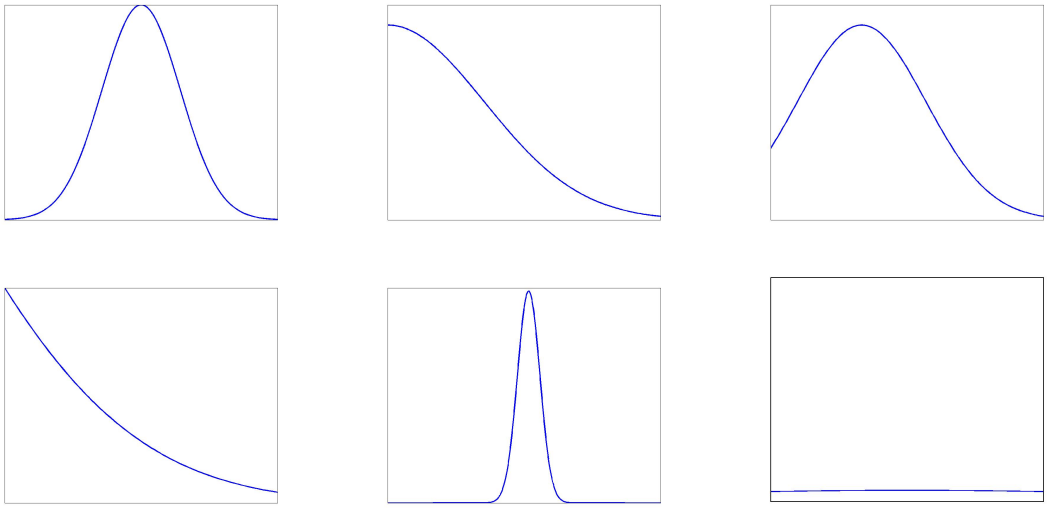

Figure 8.2: Examples for the rectified Gaussian distribution with different parameters

The rectified Gaussian distribution can model a variety of different shapes (see Fig. 8.2 for examples) For $a < 0$ and $b \neq 0$ the distribution has a Gaussian, for $a = 0$ and $b \neq 0$ an Exponential, and for $a = 0$ $b = 0$ it has a Uniform shape.
The expectations required for our purpose w.r.t. a rectified Gaussian distribution are given by

$$\langle\theta\rangle_{\mathcal{N}^+} = \mu + \frac{\sigma^2}{Z_{\mathcal{N}^+}} e^{-\frac{\mu^2}{2\sigma^2}} \tag{8.70}$$

$$\langle\theta^2\rangle_{\mathcal{N}^+} = \sigma^2 + \mu \cdot \langle\theta\rangle_{\mathcal{N}^+} \tag{8.71}$$

(see appendix C.2)

The *VBNMF* update rules

We chose the prior distributions on $\mathbf{W}$ to be rectified Gaussians

$$P(W_{ik}) \propto \left[\exp\left(-\frac{1}{2}\left(\frac{W_{ik} - \mu_{W_{0k}}}{\sigma_{W_{0k}}}\right)^2\right)\right]^+, \qquad W_{ik} \geq 0, \qquad 0 \text{ else} \tag{8.72}$$

where $[\cdot]^+$ indicates the truncation at zero and $\mu_{W_{0k}}, \sigma_{W_{0k}}$ denote the parameters of the prior distributions. One of them is assumed for each column $\mathbf{W}_{*k}$ which expresses our belief that the a priori weights of source k are assumed to be drawn from the same distribution. Similarly, we chose the prior distribution on each row $\mathbf{H}_{k*}$ to be

$$P(H_{kj}) \propto \left[\exp\left(-\frac{1}{2}\left(\frac{H_{kj} - \mu_{H_{k0}}}{\sigma_{H_{k0}}}\right)^2\right)\right]^+, \qquad H_{kj} \geq 0, \qquad 0 \text{ else} \tag{8.73}$$

Due to the conjugacy principle, it follows that

$$Q(W_{ik}) \propto \left[\exp\left(-\frac{1}{2}\left(\frac{W_{ik} - \mu_{W_{ik}}}{\sigma_{W_{ik}}}\right)^2\right)\right]^+ \tag{8.74}$$

and

$$Q(H_{kj}) \propto \left[\exp\left(-\frac{1}{2}\left(\frac{H_{kj}-\mu_{H_{kj}}}{\sigma_{H_{kj}}}\right)^2\right)\right]^+ \tag{8.75}$$

where

$$\mu_{W_{ik}} = \frac{\mu_{W_{ik}}\sigma_{W_{ik}}^{-2} + (\sigma_r)^{-2}\sum_j X_{ij}\left\langle H_{kj}\right\rangle_{Q(H_{kj})}}{\sigma_{W_{ik}}^{-2} + (\sigma_r)^{-2}\sum_j q_{ikj}^{-1}\left\langle H_{kj}^2\right\rangle_{Q(H_{kj})}} \tag{8.76}$$

$$\sigma_{W_{ik}} = \left(\sigma_{W_{ik}}^{-2} - (\sigma_r)^{-2}\sum_j q_{ikj}^{-1}\langle H_{kj}^2\rangle_{Q(H_{kj})}\right)^{-\frac{1}{2}} \tag{8.77}$$

and

$$\mu_{H_{kj}} = \frac{\mu_{H_{kj}}\sigma_{H_{kj}}^{-2} + (\sigma_r)^{-2}\sum_i X_{ij}\left\langle W_{ik}\right\rangle_{Q(W_{ik})}}{\sigma_{H_{kj}}^{-2} + (\sigma_r)^{-2}\sum_i q_{ikj}^{-1}\left\langle W_{ik}^2\right\rangle_{Q(W_{ik})}} \tag{8.78}$$

$$\sigma_{H_{kj}} = \left(\sigma_{H_{kj}}^{-2} - (\sigma_r)^{-2}\sum_i q_{ikj}^{-1}\left\langle W_{ik}^2\right\rangle_{Q(W_{ik})}\right)^{-\frac{1}{2}} \tag{8.79}$$

Equations (8.76)-(8.79) follow by quadratic complement (see appendix C.2).
If we have no knowledge on the actual values of the prior parameters, these hyperparameters $\mu_{W_{0k}}$, $\sigma_{W_{0k}}$ and $\mu_{H_{k0}}$, $\sigma_{H_{k0}}$ can also be updated by the algorithm by solving the implicit equations (8.80)-(8.83) by iteration:

$$\frac{\partial \mathcal{B}_q}{\partial \mu_{W_{0k}}} \stackrel{!}{=} 0 \qquad \Leftrightarrow \qquad \mu_{W_{0k}} = \frac{1}{N}\sum_i \langle W_{ik}\rangle_{Q(W_{ik})} - \frac{\sigma_{W_{0k}}^2}{Z_{W_k}} e^{-\frac{\mu_{W_{0k}}^2}{2\sigma_{W_{0k}}^2}} \tag{8.80}$$

$$\frac{\partial \mathcal{B}_q}{\partial \sigma_{W_{0k}}} \stackrel{!}{=} 0 \qquad \Leftrightarrow \qquad \sigma_{W_{0k}} = \left(\frac{\sum_i \left\langle (W_{ik}-\mu_{W_{0k}})^2\right\rangle_{Q(W_{ik})}}{N\left(1 - \frac{\mu_{W_{0k}}}{Z_{W_k}} e^{-\frac{\mu_{W_{0k}}^2}{2\sigma_{W_{0k}}^2}}\right)}\right)^{\frac{1}{2}} \tag{8.81}$$

and

$$\frac{\partial \mathcal{B}_q}{\partial \mu_{H_{k0}}} \stackrel{!}{=} 0 \qquad \Leftrightarrow \qquad \mu_{H_{k0}} = \frac{1}{N}\sum_j \langle H_{kj}\rangle_{Q(H_{kj})} - \frac{\sigma_{H_{k0}}^2}{Z_{H_k}} e^{-\frac{\mu_{H_{k0}}^2}{2\sigma_{H_{k0}}^2}} \tag{8.82}$$

$$\frac{\partial \mathcal{B}_q}{\partial \sigma_{H_{k0}}} \stackrel{!}{=} 0 \qquad \Leftrightarrow \qquad \sigma_{H_{k0}} = \left(\frac{\sum_j \left\langle (H_{kj}-\mu_{H_{k0}})^2\right\rangle_{Q(H_{kj})}}{M\left(1 - \frac{\mu_{H_{k0}}}{Z_{H_k}} e^{-\frac{\mu_{H_{k0}}^2}{2\sigma_{H_{k0}}^2}}\right)}\right)^{\frac{1}{2}} \tag{8.83}$$

(see appendix C.3 for the derivations).

The *VBNMF* algorithm

The overall Variational Bayes NMF (*VBNMF*) algorithm can be summarized by

- set hyperparameters $\mu_{H_{k0}}, \sigma_{H_{k0}}, \mu_{W_{0k}}, \sigma_{W_{0k}}$
- initialize $\mu_{H_{kj}}$, $\sigma_{H_{kj}}$, $\mu_{W_{ik}}$, $\sigma_{W_{ik}}$ and σ_r
- Repeat
  1. compute all q_{ikj} via eq. (8.63)
  2. update all $\mu_{H_{kj}}$, $\sigma_{H_{kj}}$ using eqns. (8.78, 8.79)
  3. update all $\mu_{W_{ik}}$ and $\sigma_{W_{ik}}$ using eqns. (8.76,8.77)
  4. update σ_r via eq. (8.64) *
  5. update hyperparameters using eqns. (8.80)-(8.83) *
  6. compute $\mathcal{B}_q$ (eq. 8.53) **

  until convergence

Steps $4-5$ * are optional if no prior knowledge exists, step 6 ** can be performed only occasionally testing convergence. The hyperparameter update equations (5) need to be iterated in sub-loops, since they have to be solved self-consistent.
Since the overall algorithm is a generalized EM algorithm (see paragraph 6.2.2 and [Att00], [GB01]) convergence towards a local optimum is assured.
To avoid getting stuck in a local optimum, the overall procedure should be repeated several times using different initializations. To save computational time, the maximum likelihood estimates gained via some standard NMF algorithm can be used to derive the necessary initial values. For example, $\sigma_{H_{kj}}$ can be approximated by a Gaussian approximation similar as in paragraph (8.1) from H_{kj}^{ML}, while $\mu_{H_{kj}}$ can be estimated if we associate $\langle H_{kj} \rangle_{\mathcal{N}^+(\mu_{H_{kj}}, \sigma_{H_{kj}})} = H_{kj}^{ML}$. Similarly, the hyperparameters $\mu_{H_{k0}}$ and $\sigma_{H_{k0}}$ can be estimated from the distributions of the whole row $\mathbf{H}_{k*}^{ML}$.

8.2.3 Simulations

Toydata with valid and violated prior assumptions

Artificial datasets were generated using the fixed 3×5 matrix

$$\mathbf{H} = \begin{pmatrix} 3 & 1 & 0 & 1 & 3 \\ 1 & 2 & 3 & 4 & 5 \\ 0 & 3 & 3 & 3 & 0 \end{pmatrix} \tag{8.84}$$

which is familiar from the simulations in chapter 3. Each row of matrix $\mathbf{W}$ was generated by a rectified Gaussian distribution with parameters set to $\mu_{0k} = 3k$ and $\sigma_{0k} = 3k$. We discuss the two cases where

1. matrix $\mathbf{W}$ obeys the prior assumptions and is generated from a rectified distribution (figure 8.3, left)
2. several entries of the matrix $\mathbf{W}$ are set to zero, such that the W_{ik} are drawn from a mixture of a rectified Gaussian and a peak at zero (figure 8.3, right)

to investigate the robustness of the technique w.r.t. violations of the prior assumptions.
The data $\mathbf{X}$ was then generated according to

$$\mathbf{X} = \mathbf{WH} + \mathcal{N}(0, \sigma_r) \tag{8.85}$$

where negative elements in $\mathbf{X}$ were set to zero. The *VBNMF* algorithm was applied while varying the factorization rank $K = 2, \ldots, 8$. We algorithmically update the hyperparameters as well, since in general this prior knowledge is not available. Note that the fixed matrix $\mathbf{H}$ given by equation (8.84) does not obey the prior distribution assumed. Since the size of $\mathbf{H} : 3 \times 5$ is small compared to the size of $\mathbf{W} : 1000 \times 3$, the violation can be ignored.

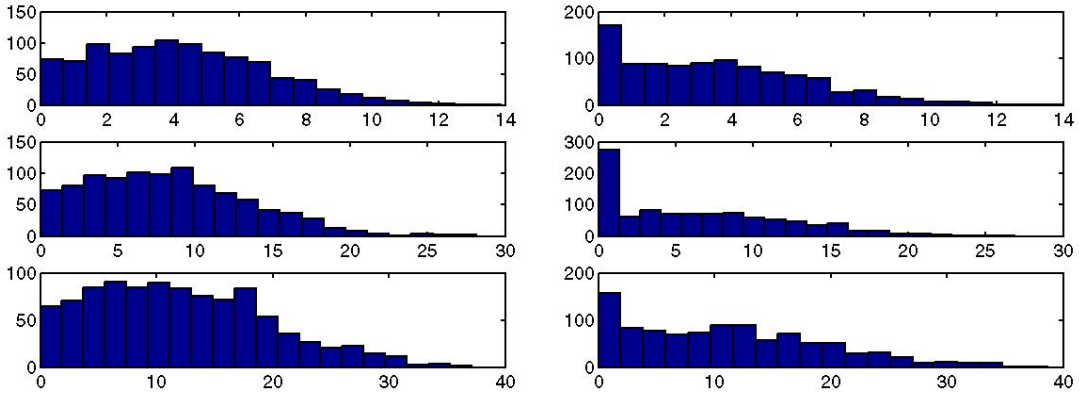

Figure 8.3: Histograms of the original columns of the 1000×3 matrix $\mathbf{W}$; *left*: rectified Gaussian $P(W_{ik}) = \mathcal{N}^+$; *right*: rectified Gaussian with additional peak at 0: $P(W_{ik}) = \tau\mathcal{N}^+ + (1-\tau)\delta(W_{ik}-0)$

Several instances of the noise parameter σ_r were tested and we eliminated possible negative data entries induced by the noise by setting them to zero. Note that this truncation constitutes a slight violation of the likelihood assumption which increases with varying noise level. If the assumption concerning the distribution of $\mathbf{W}$ holds true and the entries of each column $\mathbf{W}_{*k}$ are drawn from rectified Gaussian distribution (Fig. 8.3, left)

$$P(W_{ik}) = \mathcal{N}^+(\mu_{0k}, \sigma_{0k}) \tag{8.86}$$

the correct number of sources is detected by the *VBNMF* algorithm for various noise levels (see Fig. 8.4)
If the assumption on the prior of $\mathbf{W}$ is violated by adding a delta peak at zero to the rectified Gaussian distribution (Fig. 8.3, right)

$$P(W_{ik}) = \tau\mathcal{N}^+ + (1-\tau)\delta(W_{ik}) \tag{8.87}$$

the true number of sources $K = 3$ is detected only in the noise-free case (see Fig. 8.5). If the noise level is low ($\sigma_r = 1$ here) the log evidence bound $\mathcal{B}_q$ overestimates the number of sources slightly, while in case of high noise levels, the used model comes to a wrong answer.
The simulations consider the two cases where the assumed model either conforms to the prior assumptions or where the distribution of matrix $\mathbf{W}$ is not identical to the assumed prior form. In the first case, the *VBNMF* algorithm could indeed recover the correct number of sources by showing a clear maximum at $K = 3$ (see Fig. 8.4). In the second case, the algorithm recovered the correct K only in the absence of noise and failed at high noise levels.

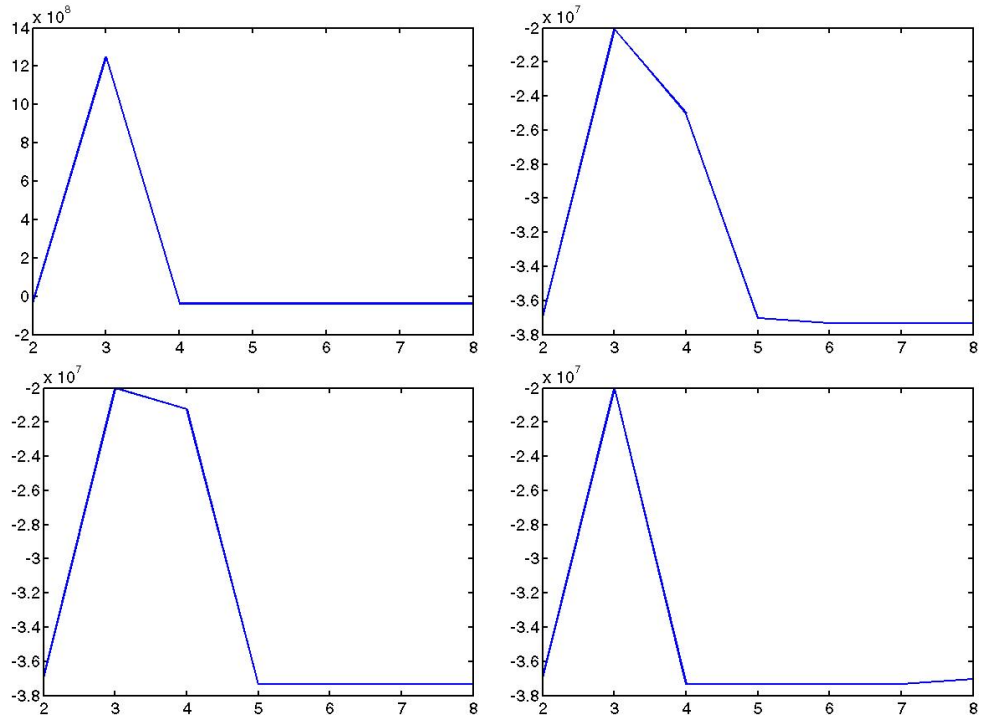

Figure 8.4: The log evidence bound $\mathcal{B}_q$ (y-axis) w.r.t. the number of components K (x-axis) for various reconstruction errors σ_r if the model assumptions hold true. *top,left*: no noise; *top,right*: $\sigma_r = 1$; *bottom,left*: $\sigma_r = 10$; *bottom,right*: $\sigma_r = 20$. The correct number of sources $K = 3$ is recognized in all cases.

Recall the individual components of the log evidence bound $\mathcal{B}_q$

$$\begin{aligned}
\mathcal{B}_q = \quad & \frac{NM}{2}\ln(2\pi) && (8.88)\\
- \quad & NM\ln\sigma_r && (8.89)\\
- \quad & \frac{1}{2\sigma_r^2}\sum_i\sum_j\{X_{ij}^2 && (8.90)\\
& -2X_{ij}\sum_k \langle W_{ik}\rangle_{Q(W_{ik})}\langle H_{kj}\rangle_{Q(H_{kj})} && (8.91)\\
& +\sum_k \frac{1}{q_{ikj}}\left\langle (W_{ik})^2\right\rangle_{Q(W_{ik})}\left\langle (H_{kj})^2\right\rangle_{Q(H_{kj})}\} && (8.92)\\
+ \quad & \sum_i\sum_k\left\langle \ln\frac{P(W_{ik})}{Q(W_{ik})}\right\rangle_{Q(W_{ik})} && (8.93)\\
+ \quad & \sum_k\sum_j\left\langle \ln\frac{P(H_{kj})}{Q(H_{kj})}\right\rangle_{Q(H_{kj})} && (8.94)
\end{aligned}$$

Terms (8.88)-(8.92) are the expected log likelihood $\langle \ln P(\mathbf{X}|\mathbf{WH},\sigma_r)\rangle_Q$ under the variational distribution Q If the number of parameters K grows, the expected maximum log likelihood grows as well

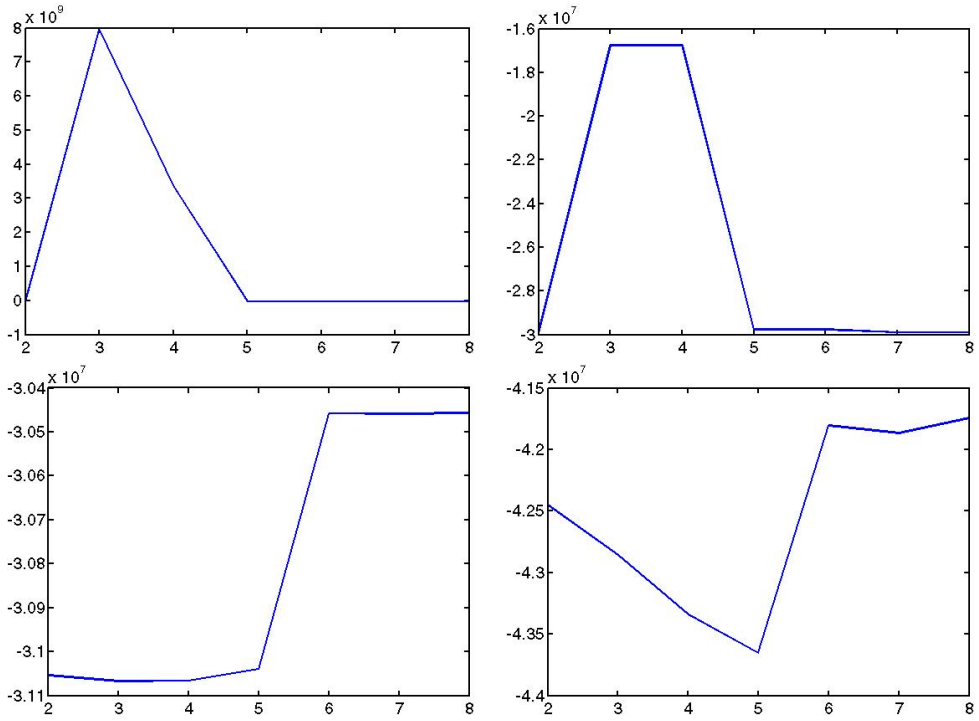

Figure 8.5: The log evidence bound $\mathcal{B}_q$ (y-axis) w.r.t. the number of components K (x-axis) for various reconstruction errors σ_r if the prior distribution assumption is violated. top,left: no noise; top,right: $\sigma_r = 1$; bottom,left: $\sigma_r = 10$; bottom,right: $\sigma_r = 20$. Only in the noise-free case the true number $K = 3$ is detected.

since the product **WH** can be more flexible adjusted to the data. To prevent the algorithm from over-fitting, a too large number of parameters must be penalized.
Term (8.93) can be written as

$$\sum_i \sum_k \int Q(W_{ik}) \ln \frac{P(W_{ik})}{Q(W_{ik})} dQ(W_{ik}) = -\sum_i \sum_k KL(Q(W_{ik})||P(W_{ik})) \tag{8.95}$$

which is the sum of negative KL-divergences $KL(Q||P)$. As a consequence of Gibb's inequality, each summand is negative (see section 6.2.1). If the number of components K is increased, the sum over k involves more terms.
Here, the parameters are bundled such that each entry of the k'th column $\mathbf{W}_{*k}$ shares the same hyperparameters $\mu_{W_{0k}}$ and $\sigma_{W_{0k}}$. Hence, with increasing K there are more individual penalty terms. If the gain in the log likelihood is small compared to the cost of extra parameters, the overall bound $\mathcal{B}_q$ will be larger if extra penalization is avoided. This can be done by setting the distributions corresponding to a whole column $Q(\mathbf{W}_{*k})$ to the prior distribution $P(W_{ik})$.
Similarly, the term (8.94) penalizes large numbers of basic patterns $\mathbf{H}_{k*}$.
The benefit of the Bayesian approach is that the balance between the log likelihood terms and the penalty terms is induced by formalism from the prior assumptions.
So, if applied to a system which fulfills all assumptions properly, a Bayesian approach will give satisfactory answers.

If some of the assumptions are violated in the system under investigation instead, the Bayesian formalism will also give an answer, but one must keep in mind that the Bayesian answer is always a consequence of the prior assumptions.
So the main focus when applying a Bayesian approach should be to carefully check the validity of the prior assumptions.

VBNMF for binary toydata

Here we investigate the potential of the *VBNMF* algorithm on a binary toydata set, similar to the one used in chapter (5).
The basic patterns given by the 4×900 matrix $\mathbf{H}$ are those familiar from the simulations on the binary NMF problem and are shown in Fig. (8.6), left. The original 1000×4 weight matrix $\mathbf{W}$ is given by columns with characteristic shapes, e.g. sinusoid or sawtooth which are easily recognized by visual inspection, but, of course unknown to the *VBNMF* algorithm (see Fig. 8.6, center).

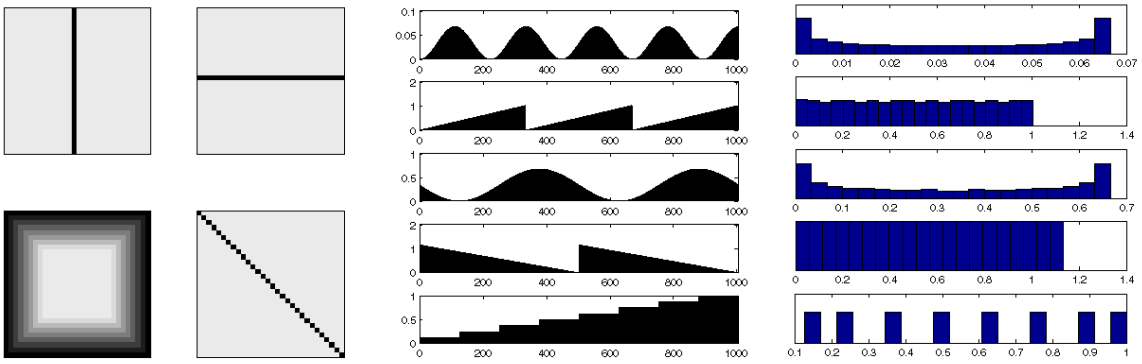

Figure 8.6: *left*: original $\mathbf{H}$; first row: $\mathbf{H}_{1*}$ $\mathbf{H}_{2*}$, second row: $\mathbf{H}_{3*}$, $\mathbf{H}_{4*}$. *center*: original weights $\mathbf{W}$; from top to bottom: $\mathbf{W}_{*1}, \ldots, \mathbf{W}_{*4}$. The last row contains weights for a constant pattern modeling noise. *right*: histograms of the columns $\mathbf{W}_{*k}$.

We added an extra component to reflect one noisy pattern by adding a row of ones to $\mathbf{H}$ and one column to $\mathbf{W}$. Alternatively, we can express the noise pattern explicitly by an N-dimensional column vector $\mathbf{c}$.
In the simulation here, the entries of $\mathbf{c}$ are gradually increasing with the row index i (see Fig. 8.6, center, last row).
The Bernoulli parameter Θ_{ij} was then generated by

$$\Theta_{ij} = 1 - \exp(-[\mathbf{WH}]_{ij} + \mathbf{c}_i) \tag{8.96}$$

and the binary data $\mathbf{X}$ was derived by setting $X_{ij} = 1$ with probability Θ_{ij} given by eq. 8.96.
We demonstrated in chapter 5 that the *binNMF* algorithm detects the correct basis patterns and weights if the actual number of components K is known.
However, the *binNMF* algorithm is a maximum likelihood technique which optimizes the Bernoulli log likelihood (8.97)

$$LL = \sum_{i=1}^{N} \sum_{j=1}^{M} \{X_{ij} \ln(1 - \exp(-[\mathbf{WH}]_{ij})) - [\mathbf{WH}]_{ij} + X_{ij}[\mathbf{WH}]_{ij}\} \tag{8.97}$$

and has no in-built mechanism which penalizes too much parameters. It would be an interesting task to create a variational Bayes analogon for the Bernoulli likelihood case. Due to the logarithm and the

non-linearity $[1 - \exp([\mathbf{WH}]_{ij})]$, the development of such an algorithm turned out to be much more complicated than for usual NMF.
However, we have seen in paragraph (5.3.4) that a rough approximation of the Bernoulli log likelihood (8.97) is given by the much simpler quadratic form

$$E(\alpha, \mathbf{W}, \mathbf{H}) = \sum_{i=1}^{N} \sum_{j=1}^{M} (\ln(1 - \alpha X_{ij}) + [\mathbf{WH}]_{ij})^2 \tag{8.98}$$

It involves an additional parameter α and leads to an NMF problem

$$\tilde{\mathbf{X}} \approx \mathbf{WH} \text{ where } \tilde{X}_{ij} = -\ln(1 - \alpha X_{ij}) \tag{8.99}$$

Here, we set $\alpha = 0.6$ which leads empirically to quite good approximations of the original data.
Obviously, the reconstruction error in this approximation

$$E_{ij} := \ln(1 - 0.6 X_{ij}) + [\mathbf{WH}]_{ij} \tag{8.100}$$

is not distributed according to a Gaussian with mean zero (see Fig. 8.7, left). which is the likelihood assumption of the *VBNMF* algorithm (see paragraph 8.2.2).
Although knowing that the assumptions on the form of the likelihood and the prior distributions are not really valid in the present example, the *VBNMF* algorithm was applied to this toydata set.
For each $K = 2 \ldots, 19$, the Alternating Least Squares (ALS) NMF algorithm as described in paragraphs (2.4.2) and (5.3.4) was run for a fixed $\alpha = 0.6$. The solutions $\mathbf{W}^{ALS}$ and $\mathbf{H}^{ALS}$ were used as initializations for $\langle \mathbf{W} \rangle_Q$ and $\langle \mathbf{H} \rangle_Q$.
The initial values of the parameters $\mu_{H_{kj}}$, $\sigma_{H_{kj}}$, $\mu_{W_{ik}}$,$\sigma_{W_{ik}}$ and σ_r and hyperparameters $\mu_{H_{k0}}$, $\sigma_{H_{k0}}$ and $\mu_{W_{0k}}$,$\sigma_{W_{0k}}$ were approximated from the ALS solutions and their statistics. This method empirically yields good initializations. An alternative, but time consuming approach would be to use multiple different random initializations, since the *VBNMF* algorithm can be stuck in local maxima.

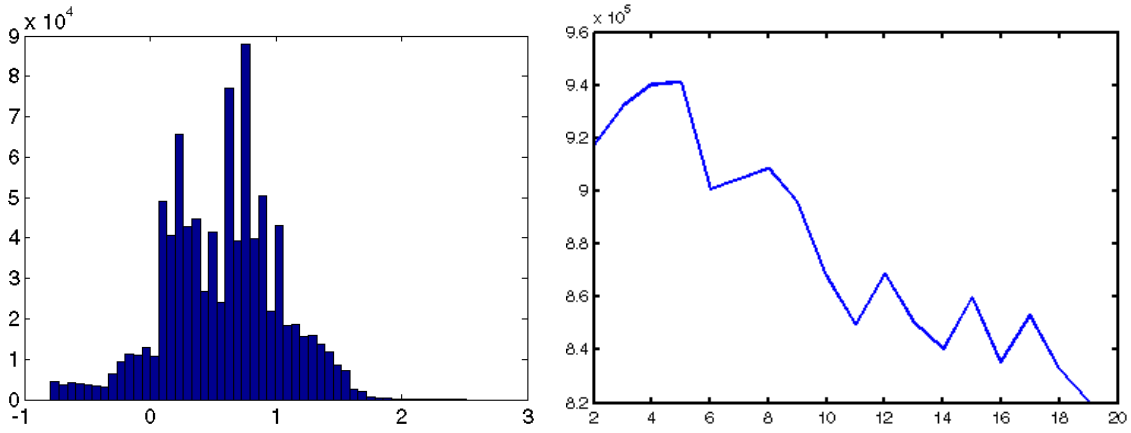

Figure 8.7: *left*: Histogram of the approximation error in the binary toydata set $E_{ij} = \ln(1-0.6X_{ij}) + [\mathbf{WH}]_{ij}$. Obviously the noise is not distributed according to a zero mean Gaussian as stated in the likelihood assumption of the *VBNMF* algorithm. *right*: The log evidence bound $\mathcal{B}$ (y-axis) w.r.t. the number of components K (x-axis) in the binary toydata example has a maximum at $K = 5$

The results gained by the *VBNMF* algorithm are surprisingly good. Even if most assumptions were violated in this toy data example, the plot of the log evidence bound (see Fig. (8.7), right) has a clear maximum at $K = 5$.

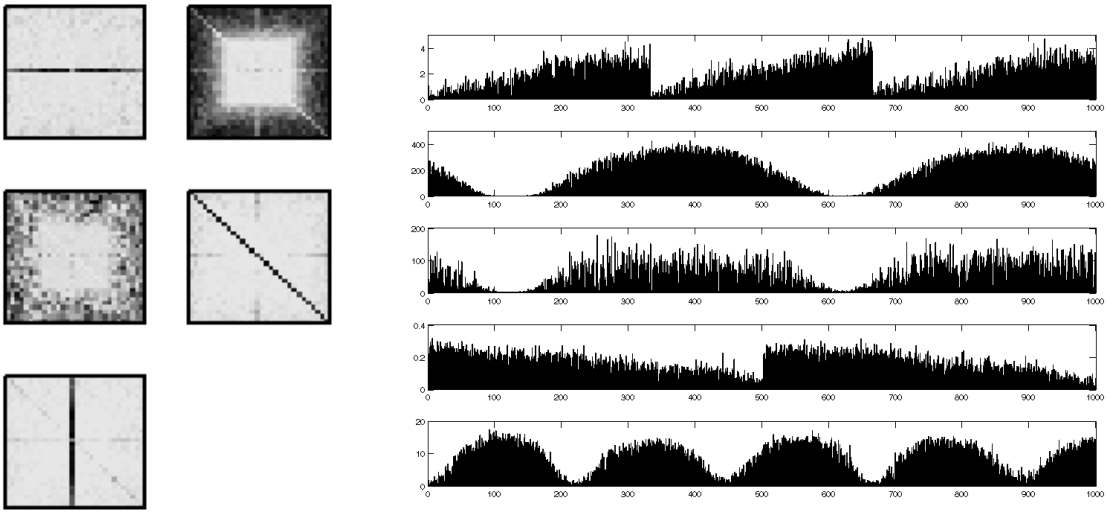

Figure 8.8: Expected basic patterns $\langle \mathbf{H} \rangle_Q$ (left) and weights $\langle \mathbf{W} \rangle_Q$ (right) using *VBNMF* and $K = 5$. The four original patterns (compare Fig. (8.6) and their weights are well approximated. The weights of an additional noise pattern differ from the original shape. One possible reason is the additional noise induced by the "binarization" of the data.

Optical inspection shows good approximations of the four original patterns plus one noise component. Even more interesting is the behavior of the *VBNMF* algorithm if K is chosen too large. In figure (8.9) an example for $K = 10$ is given. The ALS approximation decomposes the data in $K = 10$ components, some of which resemble the original patterns $\mathbf{H}_{k*}$ and weights $\mathbf{W}_{*k}$, while some of which are mixtures of these in order to fit the data $\tilde{\mathbf{X}}$ well. All available components are used in the approximation.
Initialized by the ALS solution, the *VBNMF* algorithm retains only five components while driving the remaining parameters to constants. Four of the retained components (1,5,6 and 10 in Fig. 8.9, upper right subplot) can be recognized as good approximations to the original patterns and weights. The fifth retained component (9) is interesting as well: it represents the algorithm's approximation of the remaining reconstruction error which was generated by a uniform pattern plus the noise resulting from the binary realization.
Since the algorithm has the freedom to adjust the prior hyperparameters during the optimization, the prior distributions $P(W_{ik})$ and $P(H_{kj})$ of not necessary components are driven towards narrow Gaussian peaks.
In order to avoid large penalizations, the corresponding variational distributions $Q(W_{ik})$ and $Q(H_{kj})$ are set equal to the priors. The images showing the expectations $\langle \mathbf{W} \rangle_Q$ and $\langle \mathbf{H} \rangle_Q$ show that these components are constant.
This effect is called *Automatic Relevance Determination* and was discussed by MacKay [Mac95] in a neural networks context and in [TF09] for NMF (see section 7.2.2). Note that the prior assumptions, namely the likelihood function and the parametric family of the prior distributions, as well as the independence assumptions on the prior and variational distributions determine the extent of this effect.
As this example shows, the *VBNMF* algorithm is, up to some degree, tolerant with respect to violations of these assumptions. General considerations of tolerance are an interesting topic, but will not further be discussed here.

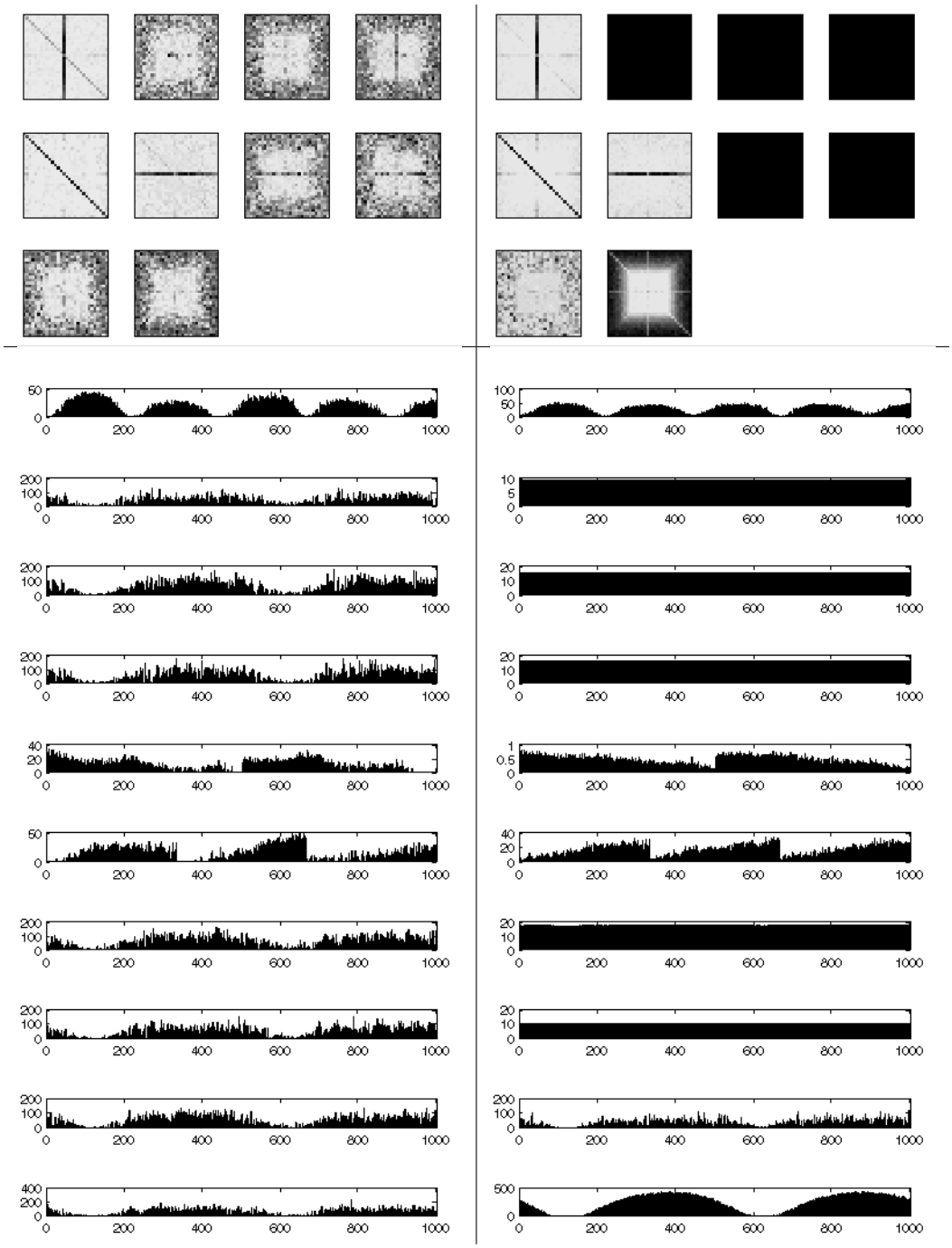

Figure 8.9: *VBNMF* in a binary toy data example where $K = 10$; *top,left*: basic patterns $\mathbf{H}^{ALS}$; *top,right*: expected basic patterns $\langle \mathbf{H} \rangle_Q$; Upper row: $\mathbf{H}_{1*}, \ldots, \mathbf{H}_{4*}$, second row: $\mathbf{H}_{5*}, \ldots, \mathbf{H}_{8*}$, third row: $\mathbf{H}_{9*}$ and $\mathbf{H}_{10*}$. *bottom,left*: weights $\mathbf{W}^{ML}$; *bottom,right*: expected weights $\langle \mathbf{W} \rangle_Q$; Only relevant components (1,5,6,9 and 10) are retained while the others are switched off. Component 9 is an attempt to explain the remaining reconstruction error.

***VBNMF* for binary real world data**

The performance of the *VBNMF* algorithm is tested on a real world data set which is a subset of 1000 wafers from the data familiar from real world example I in paragraph (5.5.1). This reduction was necessary due to memory problems when using the whole data set (for $K > 10$, especially the handling of the $N \times K \times M$ decoupling parameters q_{ikj} e.g. $> 3000 \times 10 \times 500 = 15.000.000$ for the whole data set in real world example I and the computations of the expectations $\langle\rangle_Q$ may require better implementations than the prototype used in this simulations).
The approximation given in eq. (8.99) using $\alpha = 0.6$ was used to transform the binary problem (roughly) into an NMF problem.

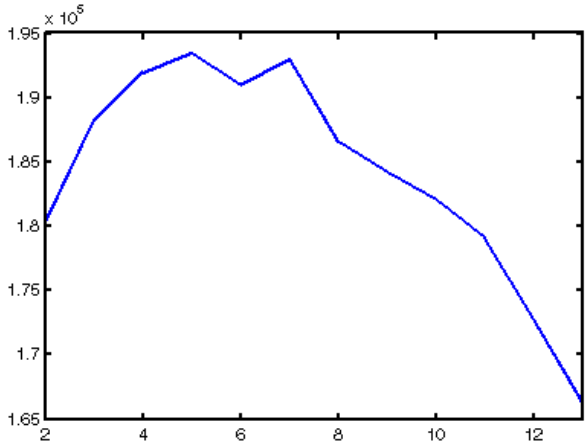

Figure 8.10: The log evidence bound $\mathcal{B}$ (y-axis) w.r.t. the number of components K (x-axis) in the binary real world example has an maximum at $K = 5$

Varying the number of components yields a maximum of the log evidence bound $\mathcal{B}_q$ at $K = 5$ (see Fig. 8.10).
Visual inspection of the expected basic patterns $\langle \mathbf{H} \rangle_Q$ for $K = 5, 6, 7$ in figure 8.11 shows that although there is some consistency, some patterns differ from each other. According to the log evidence bound, $\mathcal{B}_q(K = 5) > \mathcal{B}_q(K = 7) > \mathcal{B}_q(K = 6)$. This ranking follows by Bayesian formalism from the prior assumptions (assuming that we did every run many times and reach the global maximum for each K) and the *VBNMF* algorithm tries to explain the data by patterns $\mathbf{W}$ and $\mathbf{H}$ which are distributed according to rectified Gaussian distributions a priori. Growing numbers of parameters are penalized by the deviation of the variational distributions Q from this prior distribution P. In order to keep the penalty terms small, those components are favoured which better resemble the prior assumption. Figure 8.12 shows the approximate maximum likelihood solutions $\mathbf{W}^{ALS}$, $\mathbf{H}^{ALS}$ and the *VBNMF* expectations $\langle \mathbf{W} \rangle_Q$, $\langle \mathbf{H} \rangle_Q$ for $K = 15$. The automated relevance determination effect is obvious: only five components are retained, while the rest is driven to extinction. A comparison of the histograms of the related expected weights in the bottom of the figure shows, that the extincted components are distributed according to a narrow peak Gaussian, and the retained components are distributed according to rectified Gaussians.
In contrast, all approximate maximum likelihood counterparts gained by ALS (8.12, bottom left) have a clear peak at zero in the distribution of the weights $\mathbf{W}_{*k}$. This observation suggests that a more flexible shape for the prior distributions, such as a mixture of a rectified Gaussian and a delta peak at zero

$$P(W_{ik}) = \tau \mathcal{N}^+ + (1 - \tau)\delta(W_{ik}) \tag{8.101}$$

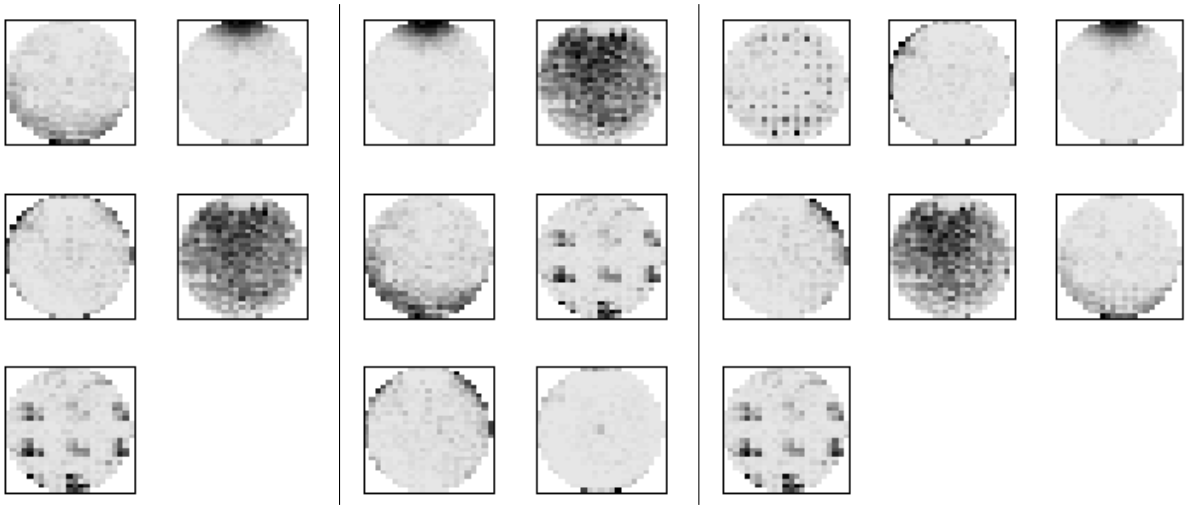

Figure 8.11: Expected basic patterns $\langle \mathbf{H} \rangle_Q$ in a real world example. All five patterns for $K = 5$ (left) appear for $K = 6$ (center) and $K = 7$ (right)

.

would better suit the underlying data and interprete more than five components to be valuable in this example. Note that priors of the form (8.101) are also conjugate to the Gaussian likelihood. The necessary computations are somewhat cumbersome, but straight forward.
Other even more flexible possible forms are mixtures of (rectified) Gaussians

$$P(W_{ik}) \propto \sum_t \alpha_t \mathcal{N}^+(\mu_t, \sigma_t), \qquad \sum_t \alpha_t = 1 \tag{8.102}$$

Note that (8.101) can be interpreted as a special case of (8.102).
In my experiments with these more flexible shapes of prior distributions, I observed that the necessary hyperparameter updates (the respective analogons to equations (8.80)-(8.83) plus updates for each parameter α_t in eq. 8.102) need more carefulness, but work in principle. Every update of a mixture distribution (8.102) requires the estimation of an optimal solution to a one-dimensional mixture-of-rectified-Gaussians model. Even usual Gaussian mixture models without truncation at zero are known to have many locally optimal solutions [Mac03]. Hence, a more sophisticated update procedure than the one used in the *VBNMF* algorithm, at least for the hyperparameters is needed.

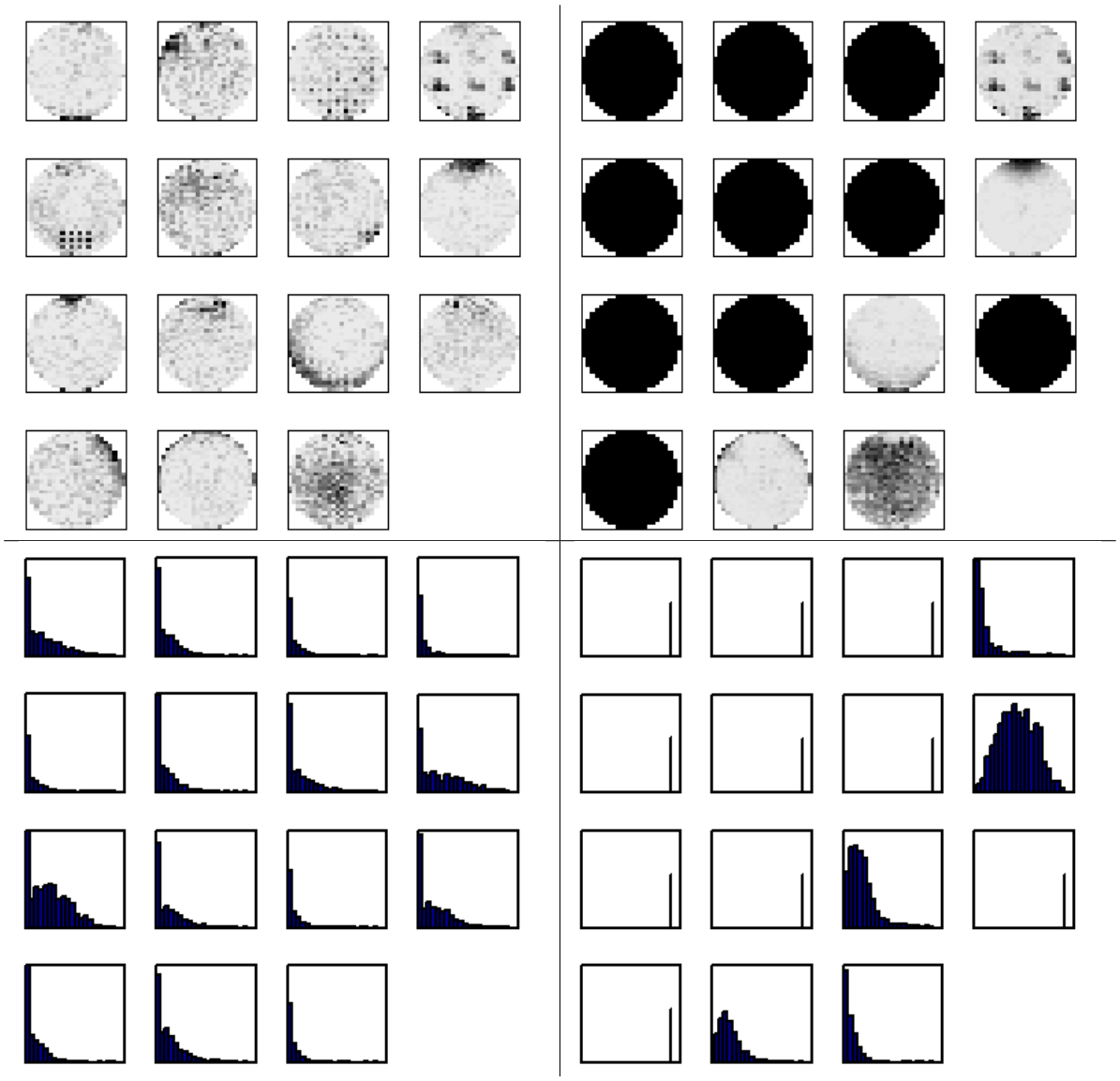

Figure 8.12: Automatic relevance determination by *VBNMF* shown in an real data example where $K = 15$ *top,left*: basic patterns $\mathbf{H}^{ALS}$ gained by approximate maximum likelihood *top,right*: expected basic patterns $\langle \mathbf{H} \rangle_Q$ gained by *VBNMF*, initialized by $\mathbf{H}^{ALS}$. *bottom,left*: histograms of weights $\mathbf{W}^{ALS}$ *bottom,right*: histograms of expected weights $\langle \mathbf{W} \rangle_Q$ Only relevant components are retained while the others are switched off by setting the respective basic pattern $\mathbf{H}_{k*}$ to a constant value and the distribution of the related weights $\mathbf{W}_{*k}$ to a narrow Gaussian peak.

8.3 Discussion

In this chapter two yet partly unsolved problems concerning NMF decompositions $\mathbf{X} \approx \mathbf{WH}$ were addressed employing Bayesian methods.
A Bayesian criterion of a *best* solution, given the correct number of components K, was obtained by integrating over all possible weight matrices $\mathbf{W}$ to estimate the posterior distribution of the basis matrix $\mathbf{H}$. In order to render the related high dimensional integral tractable, a Laplace approximation was utilized. In case of a Gaussian likelihood this method yields a simple Bayesian optimality criterion for NMF named *BOC*. It was shown that in the Gaussian likelihood case the BOC describes the same optimal solution as described by the determinant criterion in the noise-free case which was presented in chapter 3.2.4.
Further, a variational Bayes NMF algorithm for a Gaussian likelihood and rectified Gaussian priors named *VBNMF* was introduced which can be seen as a Bayesian generalization of the multiplicative Lee-Seung algorithm for Gaussian NMF. Both algorithms (the maximum likelihood procedure and its Bayesian extension) are based on Jensen's inequality to decouple parameters or expectations thereof by introducing a bound containing auxiliary variables.
Although variational Bayes procedures have already been applied previously to the NMF problem, a complete derivation for the Gaussian likelihood case with simulations has not been published so far. While [Cem09] concentrates on the KL-divergence cost function for NMF, [FC09] mentions the feasibility of the method in the Gaussian likelihood case only. Furthermore they mention that their maximum likelihood version for Gaussian NMF does not coincide with the original multiplicative NMF algorithm, while the method presented here does so.
The non-negative factor analysis based on Bayesian learning presented in [HK07] has also some parallels with the *VBNMF* method. The paper mentions that the computations with rectified Gaussian priors are only possible if their location parameter is set to zero and present some special constuction of a Gaussian prior and a rectification. Maybe the Jensen trick and the additional parameters q_{ikj} circumvent this problem in case of the *VBNMF* algorithm, since the location parameters $\mu_{H_{k0}}$ and $\mu_{W_{0k}}$ are free and the model is still tractable.
In this respect, the *VBNMF* algorithm as a direct generalization of the Lee-Seung multiplicative NMF algorithm for Euclidean NMF seems to be a new contribution to the best of my knowledge.
In simulation studies, the ability of the *VBNMF* algorithm to automatically detect the optimal number of components in constructed toydata was demonstrated. If all prior assumptions are valid, the correct number K is perfectly determined by the algorithm. Moreover, if the prior assumptions are slightly violated or even if the likelihood function differs from the model assumption, the correct factorization rank can be determined by either evaluating the lower bound to the log evidence $\mathcal{B}_q$ or by automatic relevance determination.
Decompositions of binary real world data show that the *VBNMF* algorithm in principle does distinguish between important and irrelevant components. Potential improvement of the method could be achieved by allowing more flexible shapes for the prior distributions which better suit the data. However, this improvement requires more sophisticated optimization strategies for the determination of optimal hyperparameter estimates.

Chapter 9

Summary and outlook

This thesis investigated extensions to the Non-negative matrix factorization (NMF) machine learning technique and their potential usefulness for failure analysis in semiconductor microchip manufacturing. As reviewed in the introduction (chapter 1), there is a huge arsenal of data analysis techniques used to gather information from the immense amount of data arising during and after wafer fabrications. The main idea followed here is to apply Blind Source Separation techniques, which interpret the wafer fabrication as a black box in which a set of underlying root causes somehow generates malfunctions on the chips.
NMF is suitable to discover latent structures in high dimensional data which are generated as strictly additive linear combinations of non-negative variables.
Chapter 2, gave an introduction to the NMF technique and motivated NMF problems by an air pollution example. After an overview of application fields of NMF so far, the basic principles concerning non-negatively constrained optimization are discussed. Further, two main problems are highlighted: The uniqueness problem of NMF and the lack of adequate methods to determine an optimal number of components in previously unknown data.
Before NMF was tested in a practical environment, the uniqueness issue was discussed in detail in chapter 3. By geometrical considerations the ambiguity of NMF to produce unique solutions was explained and a determinant criterion was introduced to fix this uniqueness problem. An algorithm named *detNMF* was designed which incorporates the determinant criterion directly and simulations demonstrate its ability to recover the correct solutions in constructed toydata examples. A further illustrative example was provided which distinguishes the new method from an established approach based on sparsity assumptions. Further, an alternative explanation of the apparent superiority of so called multilayer approaches for NMF could be derived by means of the determinant criterion.
In chapter 4, NMF was directly applied to a real world data set which was generated from measurement data containing different test categories called BINs. It was demonstrated that NMF clearly recovers the structure of the data. However, the determination of an optimal number of underlying components turned out to be tricky. While in this chapter, the existing NMF methodology was applied to suitable aggregated data which approximately satisfies the linear NMF model, chapter 5 presents a completely new developed extension of NMF for binary data.
The new method called *binNMF* (binary Non-negative matrix factorization) is based on a superposition approximation for fail probabilities of single chips. It turned out that these pass/fail probabilities can be modeled by means of a nonlinear function of two non-negative matrices. These parallels to usual NMF were exploited to develop an effective two stage optimization procedure which was demonstrated to discover localized features in binary test data sets.
The rest of the thesis is dedicated to the use of Bayesian techniques in combination with NMF and their ability to answer the two questions concerning the uniqueness and the optimal number of sources.

After two literature chapters on Bayesian methods in general (6), and their combination with NMF (7) further Bayesian extensions to NMF were presented in chapter 8. The idea behind these extensions was to exploit the statistical well-founded Bayesian methodology in order to find answers to the two main problems of usual NMF: uniqueness and the optimal number of components.

In the first part, a Bayesian optimality criterion was derived which describes the optimal solution of a NMF problem in absence of prior knowledge, formulated as MAP (maximum a posteriori) estimation. Unlike usual Bayesian approaches which try to explicitly model certain characteristics such as sparseness or smoothness of the components directly, here we used the Bayesian formalism to explicitly model the absence of prior information. An optimal set of basis vectors can be determined then by integrating over all possible weight matrices, given the correct factorization rank.

In the special case of a Gaussian likelihood, the Bayesian criterion can be shown to describe the same optimal solution which was proposed in chapter 3 by the determinant criterion. Unlike the usual direct implementation of certain characteristics in usual Bayes NMF approaches, here and in the determinant criterion an indirect relation between the basis components follows automatically from geometrical considerations and the Bayesian formalism.

In the second part of chapter 8, a variational Bayes algorithm named *VBNMF* was developed. It is based on the assumptions of a Gaussian likelihood and independent rectified Gaussian prior distributions on the parameter matrices. It was shown that the *VBNMF* algorithm is a direct Bayesian extension to a popular multiplicative NMF algorithm.

Simulations demonstrate that the *VBNMF* is able to discover the actual number of components in toydata sets. Remarkably, the correct number of components could also predicted well in cases where not all a priori assumptions hold true.

Application on real binary data demonstrates the general power of the method to allow conclusions on the actual number of components under certain assumptions. At the same time, the real data application shows the direction of further improvements by modeling more flexible shapes for the prior distributions.

9.1 Main contributions

To the best of my knowledge, NMF techniques have not been applied to the kind of wafer test data investigated here so far.

Separate from the exploration of this new application field for NMF and extensions, a variety of new aspects concerning the general methodology were developed here for the first time.

These findings can best be summarized by the expression *extensions to NMF*. First, the characterization of optimal solutions by a determinant criterion (chapter 3), its distinction from sparsity approaches and the simple explanation of the superiority of multilayer NMF algorithms over usual ones by means of the determinant criterion are original contributions which have been presented at the ICA conference 2009 [SPTL09].

The complete *binNMF* methodology including its optimization strategy as discussed in chapter 5 is a new extension of NMF to binary data sets. Although there are some approaches to the analysis of binary datasets in the literature, none of the existing techniques covers all aspects required for the special kind of binary pass/fail data considered here (see the discussion in paragraph 5.6). Parts of this chapter have been published GfKl conference 2008 [SPL10], and the at the ICA conference 2009 [SPL09].

Finally, the Bayesian extensions discussed in chapter 8 contain partly novel contributions.

As the discussion in paragraph 8.3 shows, the *VBNMF* algorithm as discussed here seems to be new in this form. However, there are several possibilities to develop a variational Bayes extension for NMF as has been mentioned elsewhere.

The Bayesian optimality criterion developed in section (8.1) is new and provides an alternative view on the determinant criterion.

9.2 Outlook

This thesis concludes with Bayesian approaches to determine the optimal number of components. As suggested in paragraph (8.2.3), more flexible forms for the prior distributions such as a mixture of rectified Gaussian distributions or at least a mixture of a rectified Gaussian and a delta peak at zero would better suit the real world data considered. First experiments were promising, but algorithmical problems forced my decision to end the thesis at this point and leave some room for future investigations.

Bayesian *binNMF*

Another interesting option is a variational Bayesian extension of the *binNMF* technique from chapter 5, since the quadratic approximation to the Bernoulli likelihood seems somewhat brute. The main problem herewith is to find a way to decouple the necessary expectations of the form

$$\langle \ln(1 - \exp(-[\mathbf{WH}]_{ij})) \rangle_{Q(\mathbf{W},\mathbf{H})} \tag{9.1}$$

It is not clear whether a suitable form for the variational distribution $Q(\mathbf{W}, \mathbf{H})$ can be found.
Again, a Jensen bound can be introduced [JJ99] to get the sum over k out of the nonlinear function:

$$\ln\left(1 - \exp\left(-\sum_k z_k\right)\right) = \ln\left(1 - \exp\left(-\sum_k q_k \frac{z_k}{q_k}\right)\right) \geq \sum_k q_k \ln\left(1 - \exp\left(-\frac{z_k}{q_k}\right)\right) \tag{9.2}$$

and a further possible simplification is induced by

$$\ln(1 - \exp(-x)) = -\sum_{l=0}^{\infty} \ln(1 + \exp(-2^l x)) \tag{9.3}$$

where each term on the right hand side can be approximated by a quadratic expression (see [JJ96]), but the design of an effective algorithm seems to imply further complications. For example the straight forward computation of the required update rules for the parameters q_k does not have a closed form due to the non-linearity.
Maybe, sampling techniques provide a better alternative in this case.

Tensor factorizations

Another natural extension of this work is the design of suitable tensor factorization algorithms. As mentioned in the literature survey for NMF in chapter 2, the extension of non-negative matrix factorization to the non-negative tensor factorization is an area of active current research [CZPA09].
In the applications discussed here, the data was aggregated to form a matrix containing either the wafer and chip information or the wafer and BIN category in its columns. A straight generalization thereof is a tensor containing wafer and chip and test information. (see Fig. 9.1).
The decomposition of such higher order tensors into some underlying components is expected to reveal hidden information which can not be extracted from aggregated matrices.

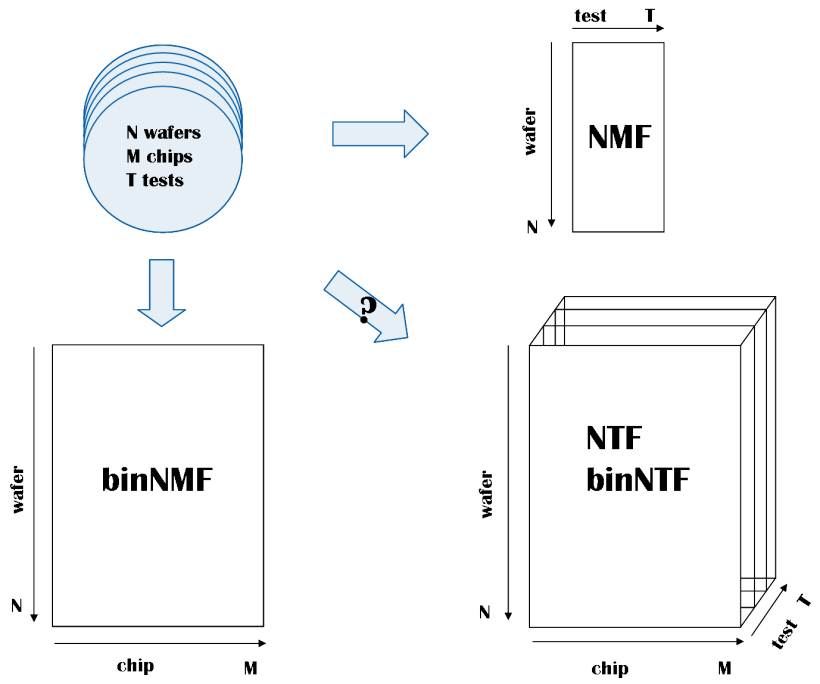

Figure 9.1: In general, wafer test data has at least three dimensions: while the wafer index $i = 1, \ldots, N$ and chip index $j = 1, \ldots, M$ can be used to determine a $N \times M$ matrix e.g. of pass/fail data which can be decomposed by the *binNMF* algorithm, a different $N \times T$ matrix containing wafer indices and different test categories can be decomposed by usual NMF. A natural extension is the decomposition of the $N \times M \times L$ tensor containing wafer, chip and test information. Non-negative tensor factorization (NTF) [CZPA09] or a tensor extension of the *binNMF* technique applied to wafer test data seem to be worth future investigations.

Appendix

A The derivative of the determinant

Here we show that the derivative of the determinant which is required for the *detNMF* algorithm in paragraph 3.2.4 is given by

$$\frac{\partial \det(\mathbf{H}\mathbf{H}^T)}{\partial H_{lm}} = 2\det(\mathbf{H}\mathbf{H}^T)\left[(\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{H}\right]_{lm} \tag{4}$$

We utilize that

$$\begin{aligned}\det(\mathbf{H}\mathbf{H}^T) &= e^{\ln(\det(\mathbf{H}\mathbf{H}^T))} = e^{\ln(\det(\mathbf{P}\mathbf{H}\mathbf{H}^T\mathbf{P}^{-1}))} \\ &= e^{\mathrm{tr}\ln(\det(\mathbf{P}\mathbf{H}\mathbf{H}^T\mathbf{P}^{-1}))} = e^{\mathrm{tr}\ln(\mathbf{H}\mathbf{H}^T)}\end{aligned}$$

where $\mathbf{P}$ is the matrix which transforms $\mathbf{H}\mathbf{H}^T$ into diagonal form $\mathbf{P}\mathbf{H}\mathbf{H}^T\mathbf{P}^{-1}$ and *tr* denotes the trace of a matrix summing over all diagonal elements. We further assume that the expression $\ln(\mathbf{H}\mathbf{H}^T)$ can be defined (in some suitable way) .
Then,

$$\begin{aligned}\frac{\partial \det(\mathbf{H}\mathbf{H}^T)}{\partial H_{lm}} &= \frac{\partial}{\partial H_{lm}} e^{\mathrm{tr}\ln(\mathbf{H}\mathbf{H}^T)} \\ &= e^{\mathrm{tr}\ln(\mathbf{H}\mathbf{H}^T)}\mathrm{tr}\frac{\partial}{\partial H_{lm}}\ln(\mathbf{H}\mathbf{H}^T) \\ &= \det(\mathbf{H}\mathbf{H}^T)\mathrm{tr}(\mathbf{H}\mathbf{H}^T)^{-1}\frac{\partial}{\partial H_{lm}}(\mathbf{H}\mathbf{H}^T) \\ &= \det(\mathbf{H}\mathbf{H}^T)\sum_i\left[(\mathbf{H}\mathbf{H}^T)^{-1}\frac{\partial}{\partial H_{lm}}(\mathbf{H}\mathbf{H}^T)\right]_{ii} \\ &= \det(\mathbf{H}\mathbf{H}^T)\sum_i\sum_j\left[(\mathbf{H}\mathbf{H}^T)^{-1}\right]_{ij}\left[\frac{\partial}{\partial H_{lm}}(\mathbf{H}\mathbf{H}^T)\right]_{ji} \\ &= \det(\mathbf{H}\mathbf{H}^T)\sum_i\sum_j\left[(\mathbf{H}\mathbf{H}^T)^{-1}\right]_{ij}(\delta_{lj}H_{im} + H_{jm}\delta_{lj}) \\ &= \det(\mathbf{H}\mathbf{H}^T)\{\sum_i\left[(\mathbf{H}\mathbf{H}^T)^{-1}\right]_{il}H_{im} + \sum_j\left[(\mathbf{H}\mathbf{H}^T)^{-1}\right]_{lj}H_{jm}\} \\ &= \det(\mathbf{H}\mathbf{H}^T)\left\{\left[\mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}\right]_{ml} + \left[(\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{H}\right]_{lm}\right\} \\ &= 2\det(\mathbf{H}\mathbf{H}^T)\left[(\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{H}\right]_{lm}\end{aligned}$$

In the last line, we used the fact that $\mathrm{pinv}(\mathbf{H}^T) = (\mathrm{pinv}(\mathbf{H}))^T$, where *pinv* denotes the Moore-Penrose pseudo-inverse of a matrix.

B Other Cost functions for the binary NMF problem

The Bernoulli likelihood as discussed in section 5.2.3 is not the only possibility to handle the binary NMF problem.
Consider the following class of cost functions :

$$E_{pq}(\mathbf{X}, \mathbf{W}, \mathbf{H}) := \sum_{i=1}^{N} \left(\sum_{j=1}^{M} \left| \mathbf{X}_{ij} - 1 + e^{-[\mathbf{WH}]_{ij}} \right|^p \right)^q, \quad p, q > 0, \tag{5}$$

This class of cost functions is related to the Minkowski-R measure [Bis96], which is used as error function e.g. in neural network applications.
Observations $\mathbf{X_{ij}} = 1$ will be approximated by large $[\mathbf{WH}]_{ij}$, while observations $\mathbf{X_{ij}} = 0$ by small $[\mathbf{WH}]_{ij}$ in order to keep the value of E_{pq} small. The individual penalty terms depend on the actual value of p and q leading to different characteristics of the optimal solutions.
In the simulations, a non-negatively constrained gradient descent algorithm in E_{pq}

$$W_{ik} \leftarrow W_{ik} - \eta_W \frac{\partial E_{pq}}{\partial W_{ik}} \tag{6}$$

$$H_{kj} \leftarrow H_{kj} - \eta_H \frac{\partial E_{pq}}{\partial H_{kj}} \tag{7}$$

was implemented.
See Fig. 2 for the evolution of the cost function E_{pq} during the first 1000 iterations in 50 random initializations on a toy dataset. The constellations $(p = 1, q = 2)$, $(p = 1, q = 3)$ and $(p = 3, q = 2)$ seem to have many local minima (in this toy data simulation). In Fig. 3, the basic patterns of the run with the smallest cost function is shown.
The three cases mentioned are local minima indeed which do not resemble the original images. The case $(p = 1, q = 1)$, does not capture the graded change from the center to the edges well. The cases $p = 2$ with the quadratic error in the second row of Fig. 3 lead to the best approximations of the original images $\mathbf{H}$ while the cases $p = 3$, $q = 1$ and $q = 3$ in the last row are slight mixtures of the original images.

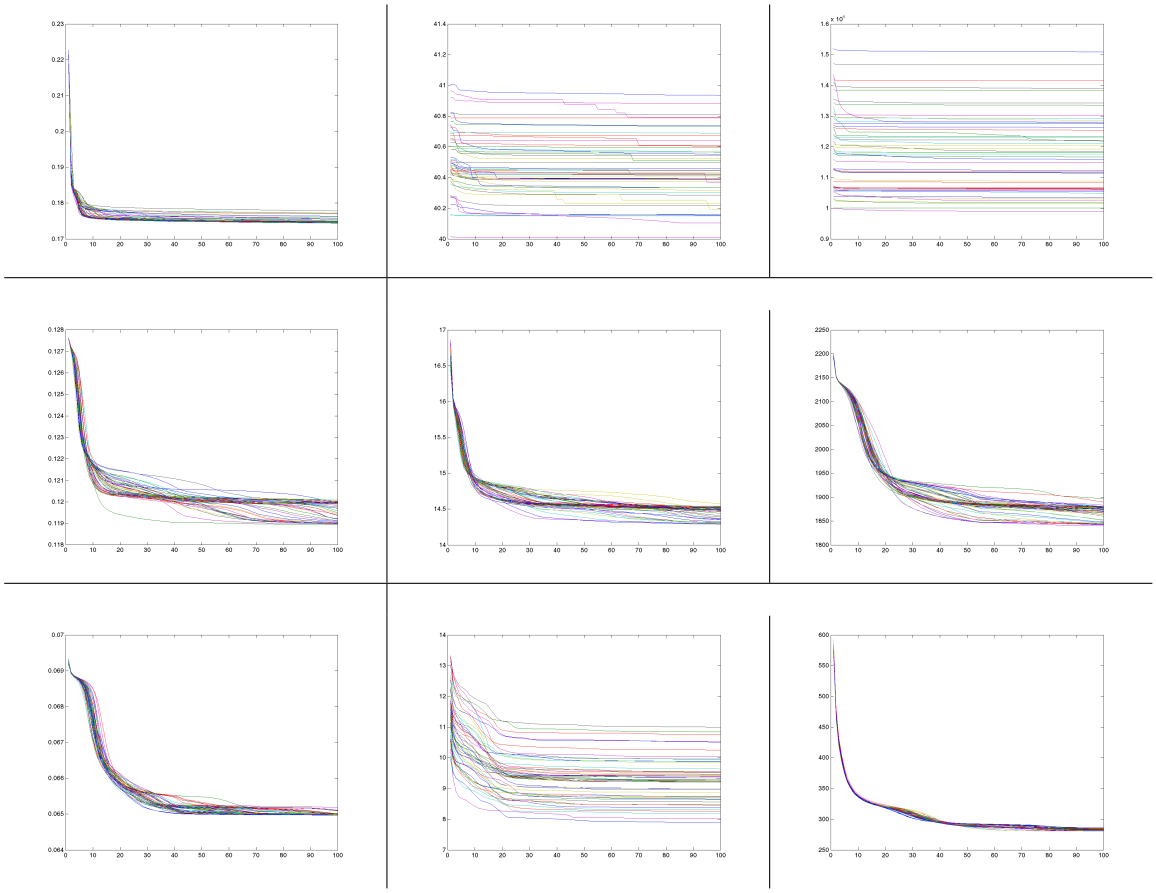

Figure 2: Evolution of the cost function E_{pq} for 50 random initializations. from top to bottom: $p = 1, 2, 3$, from left to right: $q = 1, 2, 3$.

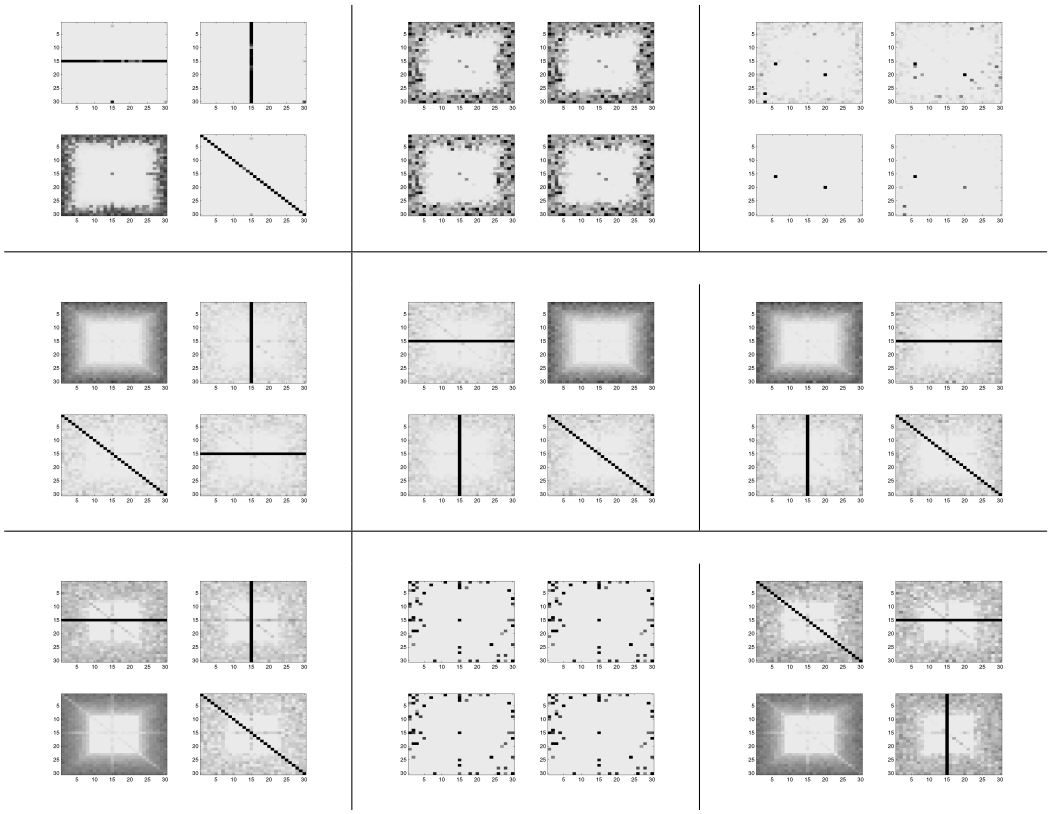

Figure 3: Examples showing the best approximations according to E_{pq} in eq. 5 among 50 randomly initialized runs. from top to bottom: $p = 1, 2, 3$, from left to right: $q = 1, 2, 3$.

C VBNMF computations

C.1 Derivation of the VBNMF updates

VBNMF update rules for $Q(H_{kj})$,$Q(W_{ik})$

Formal partial derivative of the log evidence bound $\mathcal{B}_{\mathrm{II}}$ in eq. (8.53) w.r.t the instrumental distribution $Q(H_{kj})$ under the normalization constraint yields

$$\begin{aligned}&\frac{\partial}{\partial Q(H_{kj})}\left(\mathcal{B}_q-\lambda_{kj}^H\left(\int Q(H_{kj})dQ(H_{kj})-1\right)\right)=\\&\frac{1}{2\sigma_r^2}\left\{2\sum_i X_{ij}\left\langle W_{ik}\right\rangle_{Q(W_{ik})}H_{kj}-\sum_i\frac{1}{q_{ikj}}\left\langle(W_{ik})^2\right\rangle_{Q(W_{ik})}(H_{kj})^2\right\}\\&+\ln P(H_{kj})-\ln Q(H_{kj})-1-\lambda_{kj}^H\end{aligned}$$

Setting this to zero and solving for $Q(H_{kj})$ leads to

$$Q(H_{kj})\propto P(H_{kj})\exp\left(-\frac{1}{2\sigma_r^2}\left\{\sum_i\frac{1}{q_{ikj}}\left\langle(W_{ik})^2\right\rangle_{Q(W_{ik})}(H_{kj})^2-2\sum_i X_{ij}\left\langle W_{ik}\right\rangle_{Q(W_{ik})}H_{kj}\right\}\right)$$

where the normalization constant depends on the choice of the prior distribution $P(H_{kj})$. Similarly,

$$\frac{\partial}{\partial Q(W_{ik})}\left(\mathcal{B}_q-\lambda_{ik}^W\left(\int Q(W_{ik})dQ(W_{ik})-1\right)\right)=0$$

leads to

$$Q(W_{ik})\propto P(W_{ik})\exp\left(-\frac{1}{2\sigma_r^2}\left\{\sum_j\frac{1}{q_{ikj}}\left\langle(H_{kj})^2\right\rangle_{Q(H_{kj})}(W_{ik})^2-2\sum_j X_{ij}\left\langle H_{kj}\right\rangle_{Q(H_{kj})}W_{ik}\right\}\right)$$

VBNMF update rule for q_{ikj}

$$\begin{aligned}\frac{\partial}{\partial q_{ikj}}\left(\mathcal{B}_q-\lambda_{ij}^q\left(\sum_k q_{ikj}-1\right)\right)&=&0\\\frac{1}{q_{ikj}^2}\left\langle(W_{ik})^2\right\rangle_{Q(W_{ik})}\left\langle(H_{kj})^2\right\rangle_{Q(H_{kj})}&=&2\sigma_r^2\lambda_{ij}^q\end{aligned}$$

eliminating the Lagrange multiplier λ_{ij}^q via $\sum_k q_{ikj}=1$ leads to

$$q_{ikj}=\frac{\sqrt{\left\langle(W_{ik})^2\right\rangle_{Q(W_{ik})}\left\langle(H_{kj})^2\right\rangle_{Q(H_{kj})}}}{\sum_k\sqrt{\left\langle(W_{ik})^2\right\rangle_{Q(W_{ik})}\left\langle(H_{kj})^2\right\rangle_{Q(H_{kj})}}}$$

VBNMF update rule for σ_r

The variance of the reconstruction error σ_r can be updated by solving

$$\begin{aligned}
\frac{\partial L_q}{\partial \sigma_r} &= 0 \\
\Leftrightarrow \frac{NM}{\sigma_r} &= \frac{1}{\sigma_r^3} \sum_i \sum_j \left\{ X_{ij}^2 - 2X_{ij} \sum_k \langle W_{ik} \rangle_{Q(W_{ik})} \langle H_{kj} \rangle_{Q(H_{kj})} \right. \\
&\quad \left. + \sum_k \frac{1}{q_{ikj}} \left\langle (W_{ik})^2 \right\rangle_{Q(W_{ik})} \left\langle (H_{kj})^2 \right\rangle_{Q(H_{kj})} \right\} \\
\Leftrightarrow \sigma_r^2 &= \frac{1}{NM} \sum_i \sum_j \left\{ X_{ij}^2 - 2X_{ij} \sum_k \langle W_{ik} \rangle_{Q(W_{ik})} \langle H_{kj} \rangle_{Q(H_{kj})} \right. \\
&\quad \left. + \sum_k \frac{1}{q_{ikj}} \left\langle (W_{ik})^2 \right\rangle_{Q(W_{ik})} \left\langle (H_{kj})^2 \right\rangle_{Q(H_{kj})} \right\}
\end{aligned}$$

C.2 Rectified Gaussian Computations

Gaussian distribution:

$$\mathcal{N}(\theta|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\theta-\mu}{\sigma}\right)^2}, \qquad -\infty < \theta < \infty \tag{8}$$

$$\langle \theta \rangle_{\mathcal{N}} = \mu \tag{9}$$

$$\langle \theta^2 \rangle_{\mathcal{N}} = \sigma^2 \tag{10}$$

Rectified Gaussian distribution:

The Rectified Gaussian distribution is generated from a usual Gaussian by truncating all negative values and renormalizing the integral over the positive reals to one:

$$\mathcal{N}^+(\theta|\mu,\sigma) = \frac{1}{Z_{\mathcal{N}^+}} e^{-\frac{1}{2}\left(\frac{\theta-\mu}{\sigma}\right)^2}, \qquad 0 \le \theta < \infty, \qquad 0 \text{ otherwise} \tag{11}$$

with normalization constant $Z_{\mathcal{N}^+} = \frac{1}{2}\sigma\sqrt{2\pi}\ \mathrm{erfc}\left(-\frac{\mu}{\sigma\sqrt{2}}\right)$
and $\mathrm{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_z^\infty \exp(-t^2) dt$ is the complementary error function.
The sufficient statistics for the rectified Gaussian are

$$\langle \theta \rangle_{\mathcal{N}^+} = \mu + e^{-\frac{\mu^2}{2\sigma^2}} \frac{\sqrt{2}\sigma}{\sqrt{\pi}\ \mathrm{erfc}\left(-\frac{\mu}{\sigma\sqrt{2}}\right)} = \mu + \frac{\sigma^2}{Z_{RG}} e^{-\frac{\mu^2}{2\sigma^2}} \tag{12}$$

$$\langle \theta^2 \rangle_{\mathcal{N}^+} = \sigma^2 + \mu^2 + e^{-\frac{\mu^2}{2\sigma^2}} \frac{\sqrt{2}\mu\sigma}{\sqrt{\pi}\ \mathrm{erfc}\left(-\frac{\mu}{\sigma\sqrt{2}}\right)} = \sigma^2 + \mu \langle \theta \rangle_{\mathcal{N}^+} \tag{13}$$

proof:

$$\begin{aligned}
\langle\theta\rangle_{\mathcal{N}^+} &= \frac{1}{Z_{\mathcal{N}^+}}\int_0^\infty e^{-\frac{1}{2}\left(\frac{\theta-\mu}{\sigma}\right)^2}\theta d\theta \\
&= \frac{1}{Z_{\mathcal{N}^+}}\int_{-\frac{\mu}{\sigma\sqrt{2}}}^\infty e^{-u^2}(\mu+\sqrt{2}\sigma u)\sqrt{2}\sigma du \\
&= \frac{1}{Z_{\mathcal{N}^+}}\left(\sqrt{2}\sigma\mu\int_{-\frac{\mu}{\sigma\sqrt{2}}}^\infty e^{-u^2}du + \sigma^2\int_{-\frac{\mu}{\sigma\sqrt{2}}}^\infty e^{-u^2}2udu\right) \\
&= \mu + \frac{\sigma^2}{Z_{\mathcal{N}^+}}\left[-e^{-z}\right]_{\frac{\mu^2}{2\sigma^2}}^\infty \\
&= \mu + \frac{\sigma^2}{Z_{\mathcal{N}^+}}e^{-\frac{\mu^2}{2\sigma^2}} \\
&= \mu + e^{-\frac{\mu^2}{2\sigma^2}}\frac{\sqrt{2}\sigma}{\sqrt{\pi}\ \mathrm{erfc}\left(-\frac{\mu}{\sigma\sqrt{2}}\right)}
\end{aligned}$$

$$\begin{aligned}
\langle\theta^2\rangle_{\mathcal{N}^+} &= \frac{1}{Z_{\mathcal{N}^+}}\int_0^\infty e^{-\frac{1}{2}\left(\frac{\theta-\mu}{\sigma}\right)^2}\theta^2 d\theta \\
&= \frac{1}{Z_{\mathcal{N}^+}}\left(-\sigma^2\int_0^\infty e^{-\frac{1}{2}\left(\frac{\theta-\mu}{\sigma}\right)^2}\left(-\frac{\theta-\mu}{\sigma^2}\right)\theta d\theta + \int_0^\infty e^{-\frac{1}{2}\left(\frac{\theta-\mu}{\sigma}\right)^2}\mu\theta d\theta\right) \\
&= \frac{1}{Z_{\mathcal{N}^+}}\left(-\sigma^2\left[\theta e^{-\frac{1}{2}\left(\frac{\theta-\mu}{\sigma}\right)^2}\right]_0^\infty + \sigma^2\int_0^\infty e^{-\frac{1}{2}\left(\frac{\theta-\mu}{\sigma}\right)^2}d\theta + \int_0^\infty e^{-\frac{1}{2}\left(\frac{\theta-\mu}{\sigma}\right)^2}\mu\theta d\theta\right) \\
&= \sigma^2 + \mu\langle\theta\rangle_{\mathcal{N}^+} \\
&= \sigma^2 + \mu^2 + e^{-\frac{\mu^2}{2\sigma^2}}\frac{\sqrt{2}\mu\sigma}{\sqrt{\pi}\ \mathrm{erfc}\left(-\frac{\mu}{\sigma\sqrt{2}}\right)}
\end{aligned}$$

Quadratic complement

$$
\begin{aligned}
& -\frac{1}{2}\left(\frac{\theta-\mu}{\sigma}\right)^2 + A\theta^2 + B\theta \\
= & -\left(\frac{1}{2\sigma^2} - A\right)\theta^2 + \left(\frac{\mu}{\sigma^2} + B\right)\theta - \frac{\mu^2}{2\sigma^2} \\
= & -\left(\frac{1-2\sigma^2 A}{2\sigma^2}\right)\left(\theta^2 - \frac{\mu+\sigma^2 B}{\sigma^2}\frac{2\sigma^2}{1-2\sigma^2 A}\theta\right) - \frac{\mu^2}{2\sigma^2} \\
= & -\left(\frac{\sigma^{-2}-2A}{2}\right)\left(\theta^2 - 2\frac{\mu\sigma^{-2}+B}{\sigma^{-2}-2A}\theta\right) - \frac{\mu^2}{2\sigma^2} \\
= & -\left(\frac{\sigma^{-2}-2A}{2}\right)\left(\theta - \frac{\mu\sigma^{-2}+B}{\sigma^{-2}-2A}\right)^2 + \left(\frac{\sigma^{-2}-2A}{2}\right)\left(\frac{\mu\sigma^{-2}+B}{\sigma^{-2}-2A}\right)^2 - \frac{\mu^2}{2\sigma^2} \\
= & -\frac{1}{2}\left(\underbrace{\sqrt{\frac{1}{\sigma^{-2}-2A}}}_{=:\sigma'}\right)^{-2}\left(\theta - \underbrace{\frac{\mu\sigma^{-2}+B}{\sigma^{-2}-2A}}_{=:\mu'}\right)^2 + \left(\frac{\sigma^{-2}-2A}{2}\right)\left(\frac{\mu\sigma^{-2}+B}{\sigma^{-2}-2A}\right)^2 - \frac{\mu^2}{2\sigma^2} \\
= & -\frac{1}{2}\left(\frac{\theta-\mu'}{\sigma'}\right)^2 + \frac{1}{2}\left(\frac{\mu'^2}{\sigma'^2} - \frac{\mu^2}{\sigma^2}\right)
\end{aligned}
$$

C.3 Derivation of the hyperparameter updates

Here we derive the update equations for the hyperparameters $\mu_{W_{0k}}$ and $\sigma_{W_{0k}}$ of the prior distribution for the coefficients W_{ik}

$$
P(W_{ik}|\mu_{0k}, \sigma_{W_{0k}}) = \frac{1}{Z_{W_k}} e^{-\frac{1}{2}\left(\frac{W_{ik}-\mu_{W_{0k}}}{\sigma_{W_{0k}}}\right)^2} \tag{14}
$$

where the normalization constant is $Z_{W_k} = \int_0^\infty e^{-\frac{1}{2}\left(\frac{W_{ik}-\mu_{W_{0k}}}{\sigma_{W_{0k}}}\right)^2} dW_{ik}$.
Setting the derivative of the log evidence bound $\mathcal{B}_q$ (8.53) w.r.t. the hyperparameter $\mu_{W_{0k}}$ yields

$$
\begin{aligned}
0 &\stackrel{!}{=} \frac{\partial \mathcal{B}_q}{\partial \mu_{W_{0k}}} \\
\Leftrightarrow 0 &= \frac{\partial}{\partial \mu_{W_{0k}}} \sum_i \sum_k \langle \ln(P(W_{ik}))\rangle_{Q(W_{ik})} \\
\Leftrightarrow 0 &= \frac{\partial}{\partial \mu_{W_{0k}}} \sum_i \sum_k \left\langle -\ln(Z_{W_k}) - \frac{1}{2}\left(\frac{W_{ik}-\mu_{W_{0k}}}{\sigma_{W_{0k}}}\right)^2 \right\rangle_{Q(W_{ik})} \\
\Leftrightarrow 0 &= \sum_i \left(-\frac{1}{Z_{W_k}} \frac{\partial Z_{W_k}}{\partial \mu_{W_{0k}}} + \left\langle \frac{W_{ik}-\mu_{W_{0k}}}{\sigma^2_{W_{0k}}} \right\rangle_Q \right)
\end{aligned}
$$

The partial derivative of the normalization constant Z_{W_k} w.r.t $\mu_{W_{0k}}$ is

$$\begin{aligned}
\frac{\partial Z_{W_k}}{\partial \mu_{W_{0k}}} &= \int_0^\infty e^{-\frac{1}{2}\left(\frac{W_{ik}-\mu_{W_{0k}}}{\sigma_{W_{0k}}}\right)^2} \frac{W_{ik}-\mu_{W_{0k}}}{\sigma^2_{W_{0k}}} dW_{ik} \\
&= Z_{W_k} \left\langle \frac{W_{ik}-\mu_{W_{0k}}}{\sigma^2_{W_{0k}}} \right\rangle_{P(W_{ik})} \\
&= \frac{Z_{W_k}}{\sigma^2_{W_{0k}}} \langle W_{ik} \rangle_{P(W_{ik})} - \frac{Z_{W_k}\mu_{W_{0k}}}{\sigma^2_{W_{0k}}}
\end{aligned}$$

so,

$$\begin{aligned}
\frac{\partial \mathcal{B}_q}{\partial \mu_{W_{0k}}} &= \frac{1}{\sigma^2_{W_{0k}}} \sum_i \left(\langle W_{ik} \rangle_{Q(W_{ik})} - \mu_{W_{0k}} - \langle W_{ik} \rangle_{P(W_{ik})} + \mu_{W_{0k}} \right) \\
&= \frac{1}{\sigma^2_{W_{0k}}} \sum_i \left(\langle W_{ik} \rangle_Q - \left(\mu_{W_{0k}} + \frac{\sigma^2_{W_{0k}}}{Z_{P_k}} e^{-\frac{\mu^2_{W_{0k}}}{2\sigma^2_{W_{0k}}}} \right) \right)
\end{aligned}$$

where we used eq. (12) for $\langle W_{ik} \rangle_{P(W_{ik})}$ in the second line. Setting this to zero and solving for $\mu_{W_{0k}}$ yields the implicit equation

$$\mu_{W_{0k}} = \frac{1}{N} \sum_i \langle W_{ik} \rangle_{Q(W_{ik})} - \frac{\sigma^2_{W_{0k}}}{Z_{W_k}} e^{-\frac{\mu^2_{W_{0k}}}{2\sigma^2_{W_{0k}}}} \tag{15}$$

Setting the derivative of the log evidence bound $\mathcal{B}_q$ (8.53) w.r.t. the hyperparameter $\sigma_{W_{0k}}$ yields

$$\begin{aligned}
\frac{\partial Z_{P_k}}{\partial \sigma_{W_{0k}}} &= \int_0^\infty e^{-\frac{1}{2}\left(\frac{W_{ik}-\mu_{W_{0k}}}{\sigma_{W_{0k}}}\right)^2} \frac{(W_{ik}-\mu_{W_{0k}})^2}{\sigma^3_{W_{0k}}} dW_{ik} \\
&= Z_{W_k} \left\langle \frac{(W_{ik}-\mu_{W_{0k}})^2}{\sigma^3_{W_{0k}}} \right\rangle_{P(W_{ik})} \\
&= \frac{Z_{W_k}}{\sigma^3_{W_{0k}}} \left(\langle W^2_{ik} \rangle_{P(W_{ik})} - 2\mu_{W_{0k}} \langle W_{ik} \rangle_{P(W_{ik})} + \mu^2_{W_{0k}} \right)
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial \mathcal{B}_q}{\partial \sigma_{W_{0k}}} &= \frac{1}{\sigma^3_{W_{0k}}} \sum_i \Big\{ \left\langle W_{ik}^2 \right\rangle_{Q(W_{ik})} - 2\mu_{W_{0k}} \left\langle W_{ik} \right\rangle_{Q(W_{ik})} + \mu^2_{W_{0k}} \\
&\quad - \left\langle W_{ik}^2 \right\rangle_{P(W_{ik})} + 2\mu_{W_{0k}} \left\langle W_{ik} \right\rangle_{P(W_{ik})} - \mu^2_{W_{0k}} \Big\} \\
&= \frac{1}{\sigma^3_{W_{0k}}} \sum_i \Big\{ \left\langle W_{ik}^2 \right\rangle_{Q(W_{ik})} - 2\mu_{W_{0k}} \left\langle W_{ik} \right\rangle_{Q(W_{ik})} \\
&\quad - (\sigma^2_{W_{0k}} + \mu_{W_{0k}} \left\langle W_{ik} \right\rangle_{P(W_{ik})}) + 2\mu_{W_{0k}} \left\langle W_{ik} \right\rangle_{P(W_{ik})} \Big\} \\
&= \frac{1}{\sigma^3_{W_{0k}}} \sum_i \Big\{ \left\langle W_{ik}^2 \right\rangle_{Q(W_{ik})} - 2\mu_{W_{0k}} \left\langle W_{ik} \right\rangle_{Q(W_{ik})} \\
&\quad - \sigma^2_{W_{0k}} + \mu_{W_{0k}} \left\langle W_{ik} \right\rangle_{P(W_{ik})} \Big\} \\
&= \frac{1}{\sigma^3_{W_{0k}}} \sum_i \Big\{ \left\langle W_{ik}^2 \right\rangle_{Q(W_{ik})} - 2\mu_{W_{0k}} \left\langle W_{ik} \right\rangle_{Q(W_{ik})} - \sigma^2_{W_{0k}} \\
&\quad + \mu_{W_{0k}} \left(\mu_{W_{0k}} + \frac{\sigma^2_{W_{0k}}}{Z_{W_k}} e^{-\frac{\mu^2_{W_{0k}}}{2\sigma^2_{W_{0k}}}} \right) \Big\} \\
&= \frac{1}{\sigma^3_{W_{0k}}} \sum_i \Big\{ \left\langle W_{ik}^2 \right\rangle_{Q(W_{ik})} - 2\mu_{W_{0k}} \left\langle W_{ik} \right\rangle_{Q(W_{ik})} + \mu^2_{W_{0k}} \\
&\quad + \sigma^2_{W_{0k}} \left(\frac{\mu_{W_{0k}}}{Z_{W_k}} e^{-\frac{\mu^2_{W_{0k}}}{2\sigma^2_{W_{0k}}}} - 1 \right) \Big\}
\end{aligned}$$

where we used eqs. (13) and (12). Setting this to zero and solving for $\sigma_{W_{0k}}$ yields the implicit equation

$$\sigma_{W_{0k}} = \left(\frac{\sum_i \left\langle (W_{ik} - \mu_{W_{0k}})^2 \right\rangle_{Q(W_{ik})}}{N \left(1 - \frac{\mu_{W_{0k}}}{Z_{W_k}} e^{-\frac{\mu^2_{W_{0k}}}{2\sigma^2_{W_{0k}}}} \right)} \right)^{\frac{1}{2}} \tag{16}$$

Acknowledgement

An dieser Stelle möchte ich mich ganz herzlich bei einer Reihe lieber Menschen bedanken, die mich während der vergangenen drei Jahre begleitet haben.
In erster Linie danke ich meinen beiden Betreuern, Freunden und Lehrern Gerhard Pöppel und Elmar Lang für die aussergewöhnlich angenehme Zusammenarbeit, ihr grosses Engagement und ihre Hilfe in sämtlichen Lebenslagen.
Des Weiteren danke ich allen Mitarbeitern der Firma Infineon, mit denen ich in meiner Zeit dort zu tun hatte, als da wären:
Meinem Abteilungsleiter Mathias Häuser, der mich in organisatorischen Dingen stets unterstützte, sowie allen anderen Mitarbeitern der Methodengruppe und aus anderen Abteilungen Rebecca Joyce-Woehrmann, Manfred Mirwald, Marcus Gruener, Alex Gradl, Andreas Heinrich, Stefan Gläser, Cecilie Swaenepoel, Thomas Höhme, Marielle Seitz, Lothar Pecho, André Kästner, Michael Sock, Sorin Poenariu, Peter Federl, Matthias Ernst, Michael Rettelbach, Dieter Hermans, Jens Arkenau, Bernhard Knott, Simon Jerebic, Florian Lindstaedt, Klaus Goller und Gisela Schlitt für fruchtbare Diskussionen, gute Zusammenarbeit, sowie für ihre Unterstützung und das kollegiale Arbeitsklima.
Besonderer Dank gilt Daniela Kramel von der HR Abteilung, ohne deren Vermittlungsgeschick der Kontakt zu Gerhard Pöppel zu diesem Zeitpunkt wohl nicht zustandegekommen wäre.
Danke auch den Mitgliedern der Arbeitsgruppe Lang:
Ingo Keck, Manuel Meilinger, Volker Fischer und Hans Stockmeier,
und Ana Maria Tomé für hilfreiche Diskussionen

danke Lisa, Jakob und Tuy-Anh.

Bibliography

[AHS85] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9:147–169, 1985.

[Asa05] H. Asari. Non-negative marix factorization: A possible way to learn sound dictionaries. published online at http://www.people.fas.harvard.edu/ asari/nmf.html, 2005.

[ASL09] M. Arngren, M. N. Schmidt, and J. Larsen. Bayesian nonnegative matrix factorization with volume prior for unmixing of hyperspectral images. In *Machine Learning for Signal Processing, IEEE Workshop on (MLSP)*, Sep 2009.

[Att99] H. Attias. Inferring parameters and structure of latent variable models by variational bayes. In *Proc. of the 15th Conference on Uncertainty in Artificial Intelligence*, 1999.

[Att00] H. Attias. A variational bayesian framework for graphical models. In *Advances in Neural Information Processing Systems 12*, pages 209–215. MIT Press, 2000.

[Bat96] K. J. Bathe. *Finite Element Procedures*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[BB02] G. Buchsbaum and O. Bloch. Color categories revealed by non-negative matrix factorization of munsell color spectra. *Vision Research*, 42(5):559–563, March 2002.

[BBL+06] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. In *Computational Statistics and Data Analysis*, pages 155–173, 2006.

[BBV09] N. Bertin, R. Badeau, and E. Vincent. Fast bayesian nmf algorithms enforcing harmonicity and temporal continuity in polyphonic music transcription. In *IEEE workshop on Applications of Signal Processing to Audio and Acoustics*, October 2009.

[BC99] F. Bergeret and Y. Chandon. Improving yield in ic manufacturing by statistical analysis of a large data base. *Micro*, pages 59–75, Mar. 1999.

[BCC99] E. Bertino, B. Catania, and E. Caglio. Applying data mining techniques to wafer manufacturing. In *PKDD '99: Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*, pages 41–50, London, UK, 1999. Springer-Verlag.

[Ber01] M. W. Berry, editor. *Computational information retrieval*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.

[BG04] M. J. Beal and Z. Ghahramani. Variational bayesian learning of directed graphical models with hidden variables. *Bayesian Analysis*, pages 1–44, 2004.

[Bis96] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.

[Bis99] C. M. Bishop. Variational principal components. In *Proceedings Ninth International Conference on Artificial Neural Networks, ICANN99*, pages 509–514, 1999.

[BJ06] W. Buntine and A. Jakulin. Discrete component analysis. In *LNCS*, volume 3940. Springer, 2006.

[BKF09] E. Bingham, A. Kabán, and M. Fortelius. The aspect bernoulli model: multiple causes of presences and absences. *Pattern Analysis & Applications*, 12(1):55–78, February 2009.

[BLJJ98] C. M. Bishop, N. Lawrence, T. S. Jaakkola, and M. I. Jordan. Approximating posterior distributions in belief networks using mixtures. *Advances in Neural Information Processing Systems*, 10, 1998.

[BLZ07] M. Bertero, H. Lanteri, and L. Zanni. Iterative image reconstruction: a point of view. In *Proceedings of the Interdisciplinary Workshop on Mathematical Methods in Biomedical Imaging and Intensity-Modulated Radiation Therapy (IMRT)*, October 2007.

[BNJ03] D. M. Blei, A. Y. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[BP04a] I. Buciu and I. Pitas. Application of non-negative and local non negative matrix factorization to facial expression recognition. In *ICPR (1)*, pages 288–291, 2004.

[BP04b] I. Buciu and I. Pitas. A new sparse image representation algorithm applied to facial expression recognition. In *Machine Learning for Signal Processing*, pages 539 – 548, 2004.

[BTGM04] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *PNAS*, 101(12):4164–4169, 2004.

[CA02] A. Cichocki and S.-I. Amari. *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*. Wiley and Sons, 2002.

[CC05] Y.-C. Cho and S. Choi. Nonnegative features of spectro-temporal sounds for classification. *Pattern Recogn. Lett.*, 26(9):1327–1336, 2005.

[CCR06] Z. Chen, A. Cichocki, and T. M. Rutkowski. Constrained non-negative matrix factorization method for eeg analysis in early detection of alzheimer's disease. In *IEEE International Conference on Acoustics, Speech, and Signal Processing,, ICASSP2006*, pages 893–896, 2006.

[CDPR04] M. Chu, F. Diele, R. Plemmons, and S. Ragni. Optimality, computation, and interpretation of nonnegative matrix factorizations. *SIAM JOURNAL ON MATRIX ANALYSIS*, pages 4–8030, 2004.

[CDS01] M. Collins, S. Dasgupta, and R. E. Schapire. A generalization of principal component analysis to the exponential family. *NIPS*, 11:592–598, 2001.

[Cem09] A. T. Cemgil. Bayesian inference in non-negative matrix factorisation models. *Computational Intelligence and Neuroscience*, 2009.

[CGLZ01] X. Chen, L. Gu, S. Z. Li, and H. Zhang. Learning representative local features for face detection. In *CVPR (1)*, pages 1126–1131, 2001.

[CHCL09] C.-C. Chiu, S.-Y. Hwang, D. Cook, and Y.-P. Luh. Process disturbance identification through integration of spatiotemporal ica and cart approach. *Neural Computing & Applications*, December 2009.

[Chi95] S. Chib. Marginal likelihood from the gibbs output. *Journal of the American Statistical Association*, 90:1313–1321, 1995.

[CLKC08] A. Cichocki, H.Y. Lee, Y.D. Kim, and S. Choi. Non-negative matrix factorization with alpha-divergence. *PRL*, 29(9):1433–1440, July 2008.

[Com94] P. Comon. Independent component analysis, a new concept? *Signal Process.*, 36(3):287–314, 1994.

[CP05] M. Chu and R. Plemmons. Nonnegative matrix factorization and applications. *BULLETIN OF THE INTERNATIONAL LINEAR ALGEBRA SOCIETY*, 34:2–7, 2005.

[CP07] D. Chen and R. Plemmons. Nonnegativity constraints in numerical analysis. In *Conference Proceedings of the Symposium on the Birth of Numerical Analysis, October 2007*, 2007.

[Cra94] M. Craig. Minimum volume transforms for remotely sensed data. *IEEE Trans. Geoscience and Remote Sensing*, 32(3):542–552, 1994.

[CSPMT+06] P. Carmona-Saez, R. D. Pascual-Marqui, F. Tirado, J. M. Carazo, and A. D. Pascual-Montano. Biclustering of gene expression data by non-smooth non-negative matrix factorization. *BMC Bioinformatics*, 7:78, 2006.

[CT91] T. Cover and J. Thomas. *Elements of Information Theory*. New York: John Wiley, 1991.

[CWC07] C.-F. Chien, W.-C. Wanga, and J.-C. Chenga. Data mining for yield enhancement in semiconductor manufacturing and an empirical study. *Expert Systems with Applications*, 33:192–198, July 2007.

[CZ06] A. Cichocki and R. Zdunek. Nmflab for signal and image processing. available online at *http://www.bsp.brain.riken.jp/ICALAB/nmflab.html*, 2006.

[CZ07] A. Cichocki and R. Zdunek. Multilayer nonnegative matrix factorization using projected gradient approaches. *Int. J. Neural Syst.*, 17(6):431–446, 2007.

[CZA06] A. Cichocki, R. Zdunek, and S.-I. Amari. Csiszar's divergences for non-negative matrix factorization: Family of new algorithms. In *LNCS*, volume 3889, pages 32–39. Springer, 2006.

[CZA07] A. Cichocki, R. Zdunek, and S.-I. Amari. Novel multi-layer nonnegative tensor factorization with sparsity constraints. In *LNCS*, volume 4432, pages 271–280, 2007.

[CZA08] A. Cichocki, R. Zdunek, and S.-I. Amari. Nonnegative matrix and tensor factorization. *IEEE Signal Processing Magazine*, pages 142–145, January 2008.

[CZPA09] A. Cichocki, R. Zdunek, A.-H. Phan, and S.-I. Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis*. John Wiley, November 2009.

[D.91] Montgomery D. *Introduction to Statistical Quality Control*. John Wiley & Sons, New York, 1991.

[Dev05] K. Devarajan. Molecular pattern discovery using non-negative matrix factorization based on renyi's information measure. In *FIMXII-SCMA2005@AUBURN, Twelfth Annual International Conference on Statistics, Combinatorics, Mathematics and Applications*, 2005.

[Dev08] K. Devarajan. Nonnegative matrix factorization: An analytical and interpretive tool in computational biology. *PLoS Comput Biol*, 4(7):e1000029, 07 2008.

[DLJar] C. Ding, T. Li, and M. Jordan. Convex and semi-nonnegative matrix factorizations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2010, to appear.

[DLP08] C. Ding, T. Li, and W. Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Comput. Stat. Data Anal.*, 52(8):3913–3927, 2008.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[DMSS07] S. Du, X. Mao, P. Sajda, and D. C. Shungu. Automated tissue segmentation and blind recovery of 1h mrs imaging spectral patterns of normal and diseased human brain. *NMR in Biomedicine*, 2007.

[DP87] A. R. De Pierro. On the convergence of the iterative image space reconstruction algorithm for volume ect. *IEEE TRANS. MED. IMAG.*, MI-6(2):174–175, 1987.

[DPGK05] Y. Dupret, E. Perrin, J.-L. Grolier, and R. Kielbasa. Comparison of three different methods to model the semiconductor manufacturing yield. In *Advanced Semiconductor Manufacturing Conference and Workshop*, pages 118 – 123, April 2005.

[DRdFC07] K. Drakakis, S. Rickard, R. de Fréin, and A. Cichocki. Analysis of financial data using non-negative matrix factorization. *International Journal of Mathematical Sciences, Vol. 6, No. 2*, 2007.

[DS04] D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *NIPS 2003*. MIT Press, 2004.

[DS05] I. S. Dhillon and S. Sra. Generalized nonnegative matrix approximations with bregman divergences. In *Neural Information Processing Systems*, pages 283–290. MIT Press, 2005.

[DWM86] M. E. Daube-Witherspoon and G. Muehllehner. An iterative image space reconstruction algorthm suitable for volume ect. *IEEE Transactions on Medical Imaging*, 5:61–66, June 1986.

[DZ95] P. Dayan and R. Zemel. Competition and multiple cause models. *Neural Computation*, 7(3):565–579, May 1995.

[FBD09] C. Févotte, N. Bertin, and J.-L. Durrieu. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural Comput.*, 21(3):793–830, 2009.

[FC09] C. Févotte and A. T. Cemgil. Nonnegative matrix factorisations as probabilistic inference in composite models. In *Proc. 17th European Signal Processing Conference (EUSIPCO'09)*, Glasgow, 2009.

[Fed96] P. Federl. Apparatus and method for detecting and assessing a spatially discrete dot pattern, 4 1996. US patent US5950181.

[Fey72] R. Feynman. *Statistical mechanics*. W.A. Benjamin, Reading, Mass. :, 1972.

[Fie94] D. J. Field. What is the goal of sensory coding? *Neural Comput.*, 6(4):559–601, 1994.

[FPW97] P. Federl, G. Poeppel, and F. Wachtmeister. Verfahren zur Überwachung von bearbeitungsanlagen, 12 1997. European patent EP1038163.

[GB00a] M. Gardner and J. Bieker. Data mining solves tough semiconductor manufacturing problems. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 376–383, New York, NY, USA, 2000.

[GB00b] Z. Ghahramani and M. J. Beal. Variational inference for bayesian mixtures of factor analysers. In *Advances in Neural Information Processing Systems 12*. MIT Press, 2000.

[GB01] Z. Ghahramani and M. J. Beal. Propagation algorithms for variational bayesian learning. In *In Advances in Neural Information Processing Systems 13*, pages 507–513. MIT Press, 2001.

[GC05] Y. Gao and G. Church. Improving molecular cancer class discovery through sparse non-negative matrix factorization. *Bioinformatics*, 21(21):3970–3975, 2005.

[GG87] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. In *Readings in computer vision: issues, problems, principles, and paradigms*, pages 564–584. Morgan Kaufmann Publishers Inc., 1987.

[GG05] E. Gaussier and C. Goutte. Relation between plsa and nmf and implications. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 601–602, New York, NY, USA, 2005. ACM.

[Gha04] Z. Ghahramani. Unsupervised learning. In *Advanced Lectures on Machine Learning*, pages 72–112. Springer-Verlag, 2004.

[Gor74] R. L. Gorsuch. *Factor analysis*. Saunders, Philadelphia,, 1974.

[GPH04] C. Gobinet, E. Perrin, and R. Huez. Application of nonnegative matrix factorization to fluorescence spectroscopy. In *Proc. EUSIPCO 2004*, Sept. 2004.

[GS00] L. Grippo and M. Sciandrone. On the convergence of the block nonlinear gauss-seidel method under convex constraints. *Oper. Res. Lett.*, 26(3):127–136, 2000.

[GS07] Z. Ge and Z. Song. Process monitoring based on independent component analysis and principal component analysis (ica,pca) and similarity factors. *Industrial & Engineering Chemistry Research*, 46:2054–2063, 2007.

[GV03] D. Guillamet and J. Vitrià. Evaluation of distance metrics for recognition based on non-negative matrix factorization. *Pattern Recognition Letters*, 24:1599–1605, June 2003.

[GVS03] D. Guillamet, J. Vitrià, and B. Schiele. Introducing a weighted non-negative matrix factorization for image classification. *Pattern Recogn. Lett.*, 24(14):2447–2454, 2003.

[GZ05] E. F. Gonzalez and Y. Zhang. Accelerating the lee-seung algorithm for nonnegative matrix factorization. Technical report, Department of Computational and Applied Mathematics, Rice University Houston, TX 77005, 2005.

[Hec98] D. Heckerman. A tutorial on learning with bayesian networks. In *Learning in graphical models*. Cambridge MA, MIT Press, 1998.

[HK07] M. Harva and A. Kabán. Variational learning for rectified factor analysis. *Signal Processing*, 87(3):509–527, 2007.

[HKO01] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley and Sons, 2001.

[Hof99] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, New York, NY, USA, 1999. ACM.

[Hof01] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, 2001.

[Hop00] P. K. Hopke. A guide to positive matrix factorization. available online at *http://www.epa.gov/ttnamti1/files/ambient/pm25/workshop/laymen.pdf*, 2000.

[Hot33] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[Hoy02] P. Hoyer. Non-negative sparse coding. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, pages 557–565, 2002.

[Hoy04] P. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5, 2004.

[HS86] G. E. Hinton and T. Sejnowski. Learning and relearning in boltzmann machines. *Parrallel distributed Processing*, 1, 1986.

[HvC93] G. E. Hinton and D. van Camp. Keeping neural networks simple by minimizing the sescription length of the weights. In *6th annual workshop on Computational Learning Theory*, 1993.

[HZ94] G. E. Hinton and R. S. Zemel. Autoencoders, minimum description length and helmholtz free energy. In *Advances in Neural Information Processing Systems 6*, 1994.

[Jay03] E. T. Jaynes. *Probability Theory: The Logic of*. Kluwer Academic Publishers, 2003.

[Jen06] J. L. W. V. Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Math.*, 30:175–193, 1906.

[JGJS98] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical methods. In *Machine Learning*, pages 183–233. MIT Press, 1998.

[JJ96] T. S. Jaakkola and M. I. Jordan. Computing upper and lower bounds on likelihoods in intractable networks. In *12th conference on Uncertainty and Artificial Intelligence*, pages 340–348. Morgan Kaufmann, 1996.

[JJ97] T. S. Jaakkola and M. I. Jordan. Bayesian logistic regression, a variational approach. In *Conference on Artificial Intelligence and Statistics*, 1997.

[JJ98] T. S. Jaakkola and M. I. Jordan. Improving the mean field approximation via the use of mixture distributions. In *Learning in graphical models*. Cambridge MA, MIT Press, 1998.

[JJ99] T. S. Jaakkola and M. I. Jordan. Variational probabilistic inference and the qmr-dt network. *Journal of Artificial Intelligence Research*, 10:291–322, 1999.

[Jol86] I. Jolliffe. *Principal Component Analysis.* New York: Springer-Verlag, 1986.

[Kai58] H. F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23:187–200, 1958.

[KB08] A. Kabán and E. Bingham. Factorisation and denoising of 0-1 data: A variational approach. *Neurocomputing*, 71(10-12):2291–2308, 2008.

[KBH04] A. Kabán, E. Bingham, and T. Hirsimäki. Learning to read between the lines: The aspect bernoulli model. In *4th SIAM International Conference on Data Mining*, pages 462–466, 2004.

[Kom07] R. Kompass. A generalized divergence measure for nonnegative matrix factorization. *Neural Comput.*, 19(3):780–791, 2007.

[KP07] H. Kim and H. Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, June 2007.

[KP08] H. Kim and H. Park. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM Journal on Matrix Analysis and Applications*, 30(2):713–730, 2008.

[KSD06] D. Kim, S. Sra, and I. S. Dhillon. A new projected quasi-newton approach for the nonnegative least squares problem. Technical report, UTCS Technical Report TR-06-54, 2006.

[KSD07] D. Kim, S. Sra, and I. S. Dhillon. Fast newton-type methods for the least squares nonnegative matrix approximation problem. In *in Data Mining, Proceedings of SIAM Conference on*, pages 343–354, 2007.

[KSD08] D. Kim, S. Sra, and I. S. Dhillon. Fast projection-based methods for the least squares nonnegative matrix approximation problem. *Stat. Anal. Data Min.*, 1(1):38–51, 2008.

[KWL+97] B. Koppenhoefer, S. Wuerthner, L. Ludwig, W. Rosenstiel, H.-H. Kuge, M. Hummel, and P. Federl. Analysis of electrical test data using a neural network approach. *IEEE /SEMI Advanced Semiconductor Manufacturing Conference and Workshop*, pages 37–41, 1997.

[Lap99] H. Lappalainen. Ensemble learning for independent component analysis. In *ICA'99: 1st workshop on independent component analysis and blind signal separation*, 1999.

[LC84] K. Lange and R. Carson. Em reconstruction algorithms for emission and transmission tomography. *Journal of Computer Assisted Tomography*, 8(2):306–316, April 1984.

[LCP+08] H. Laurberg, M. G. Christensen, M. D. Plumbley, L. K. Hansen, and S. H. Jensen. Theorems on positive data: On the uniqueness of nmf. *Comput. Intell. Neuroscience*, 2008.

[LH74] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. PrenticeHall, Englewood Cliffs, NJ, USA, 1974.

[LH87] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems (Classics in Applied Mathematics)*. Society for Industrial Mathematics, new edition edition, 1987.

[LH07] H. Laurberg and L. K. Hansen. On affine non-negative matrix factorization. In *ICASSP 2007, Hawaii, USA*, 2007.

[LHZC03] S. Z. Li, X. W. Hou, H. J. Zhang, and Q. S. Cheng. Learning spatially localized, parts-based representation. In *Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–207–I–212, April 2003.

[Lin07a] C.-J. Lin. On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *Neural Networks, IEEE Transactions on*, 18(6):1589–1596, 2007.

[Lin07b] C.-J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Comput.*, 19(10):2756–2779, October 2007.

[LLCL01] J. S. Lee, D. D. Lee, S. Choi, and D. S. Lee. Application of non-negative matrix factorization to dynamic positron emmision tomography. In *Proceedings of the ICA*, pages 629–632, 2001.

[LMA06] A. Langville, C. Meyer, and R. Albright. Initializations for the nonnegative matrix factorization. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 2006*, 2006.

[LQL06] J.-M. Lee, S. J. Qin, and I.-B. Lee. Fault detection and diagnosis based on modified independent component analysis. *AIChE Journal*, 52(10):3501–3514, 2006.

[LS99] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999.

[LS01] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, volume 13, pages 556–562, 2001.

[Luc74] L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79(6):745+, June 1974.

[LZL04] W. Liu, N. Zheng, and X. Li. Nonnegative matrix factorization for eeg signal classification. In *ISNN (2)*, pages 470–475, 2004.

[Mac92a] D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

[Mac92b] D. J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

[Mac95] D. J. C. MacKay. Ensemble learning and evidence maximization. Technical report, Cavendish Laboratory, University of Cambridge, 1995.

[Mac99] D. J. C. MacKay. Comparison of approximate methods for handling hyperparameters. *Neural Computation*, 11:1035–1068, 1999.

[Mac03] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. Available from `http://www.inference.phy.cam.ac.uk/mackay/itila/`.

[MBCMD04] S. Moussaoui, D. Brie, O. Caspary, and A. Mohammad-Djafari. A bayesian method for positive source separation. In *in proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2004)*, pages 485–488, 2004.

[MBMDC06] S. Moussaoui, D. Brie, A. Mohammad-Djafari, and C. Carteret. Separation of non-negative mixture of non-negative sources using a bayesian approach and mcmc sampling. *IEEE transactions on Signal Processing*, 54(11), 2006.

[MGNR06] E. Meeds, Z. Ghahramani, R. Neal, and S. Roweis. Modeling dyadic data with binary latent features. *NIPS*, 19, 2006.

[Min98] T. Minka. Expectation-maximization as lower bound maximization. Tutorial published on the web at http://www-white.media.mit.edu/tpminka/papers/em.html., 1998.

[Mit94] H. Mitter. *Quantentheorie*. B.I. Wissenschaftsverlag, 1994.

[Moo20] E. H. Moore. On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society*, 26:394–395, 1920.

[MQ07] L. Miao and H. Qi. Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. *Geoscience and Remote Sensing, IEEE Transactions on*, 45:765 – 777, March 2007.

[MZ05] M. Merritt and Y. Zhang. Interior-point gradient method for large-scale totally nonnegative least squares problems. *Journal of Optimization Theory and Applications*, 126(1):191–202, 2005.

[Nea01] R. Neal. Annealed importance sampling. *Statistics and Computing*, 11:125–139, 2001.

[NH98] R. Neal and G. E. Hinton. A view of the em algorithm that justifies incremental, sparse and other variants. In *Learning in Graphical models*. Cambridge, MIT Press, 1998.

[NM01] M. Novak and R. Mammone. Use of non-negative matrix factorization for language model adaptation in a lecture transcription task. In *In Proc. of ICASSP*, pages 541–544, 2001.

[OP06] O. Okun and H. Priisalu. Fast nonnegative matrix factorization and its application for protein fold recognition. *EURASIP J. Appl. Signal Process.*, 2006.

[OSAMB99] M. F. Ochs, R. S. Stoyanova, F. Arias-Mendoza, and T. R. Brown. A new method for spectral decomposition using a bilinear bayesian approach. *Journal of Magnetic Resonance*, 137:161–176, March 1999.

[Par88] G. Parisi. *Statistical Field Theory*. Redwood City, CA: Addison-Wesley, 1988.

[Pea01] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

[Pen55] R. Penrose. A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society*, 51:406–413, 1955.

[Pha06] H. Pham. *Springer Handbook of Engineering Statistics*. Springer, London, 2006.

[Plu01] M. Plumbley. Adaptive lateral inhibition for non-negative ica. In *ICA '01, Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation (ICA 2001)*, 2001.

[Plu02] M. Plumbley. Conditions for nonnegative independent component analysis. *IEEE Signal Processing Letters*, 9(6):177–180, 2002.

[Plu03] M. Plumbley. Algorithms for nonnegative independent component analysis. *IEEE Transactions on Neural Networks*, 14:534–543, 2003.

[PMCK+06] A. Pascual-Montano, J.M. Carazo, K. Kochi, D. Lehmann, and R.D. Pascual Marqui. Nonsmooth nonnegative matrix factorization (nsnmf). *PAMI*, 28(3):403–415, March 2006.

[PT94] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[Ric72] H. W. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62(1):55–59, January 1972.

[RMRM08] E. Ragnoli, S. McLoone, J. Ringwood, and N. Macgerailt. Matrix factorisation techniques for endpoint detection in plasma etching. In *Advanced Semiconductor Manufacturing Conference*, pages 156 – 161, May 2008.

[RPW06] T. Rohatsch, G. Pöppel, and H. Werner. Projection pursuit for analyzing data from semiconductor environments. *Semiconductor Manufacturing, IEEE Transactions on*, 19:87 – 94, Feb. 2006.

[RS61] H. Raiffa and R. Schlaifer. *Applied statistical decision theory.* MIT Press, 1961.

[Rus76] J. Rustagi. *Variational Methods in Statistics.* New York: Academic Press, 1976.

[Sau95] E. Saund. A multiple cause mixture model for unsupervised learning. *Neural Computation*, 7(1):51–71, Jan. 1995.

[SB03] P. Smaragdis and J. C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *In IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180, 2003.

[SBPP06] F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons. Document clustering using nonnegative matrix factorization. *Inf. Process. Manage.*, 42(2):373–386, 2006.

[Sch78] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[Sch02] A. Schels. Neuronale netzwerkmodelle zur analyse hochdimensionaler, multisensorischer datensätze prozessierter si-wafer, Januar 2002.

[SD06] S. Sra and I. S. Dhillon. Nonnegative matrix approximation: algorithms and applications. Technical report, UTCS Report TR-06-27, 2006.

[SDB+03] P. Sajda, S. Du, T. Brown, L. Parra, and R. Stoyanova. Recovery of constituent spectra in 3d chemical shift imaging using non-negative matrix fatorization. In *4th International Symposium on Independent Component Analysis and Blind Signal Separation*, pages 71–76, 2003.

[SDB+04] P. Sajda, S. Du, T.R. Brown, R.S. Stoyanova, D.C. Shungu, X. Mao, and L.C. Parra. Nonnegative matrix factorization for rapid recovery of constituent spectra in magnetic resonance chemical shift imaging of the brain. *MedImg*, 23(12):1453–1465, December 2004.

[SH06] T. Singliar and M. Hauskrecht. Noisy-or component analysis and its application to link analysis. *Journal of Machine Learning Research*, 7:2189–2213, 2006.

[SHLL05] J. Soenjaya, W. Hsu, M. L. Lee, and T. Lee. Mining wafer fabrication: framework and challenges. In M.M. Kantardzic and J. Zurada, editors, *Next Generation of Data-Mining Application*, pages 17–40. John Wiley & Sons, New York, 2005.

[SJ98] L. K. Saul and M. I. Jordan. A mean field learning algorithm for unsupervised neural networks. In *Learning in graphical models*. Cambridge MA, MIT Press, 1998.

[SL08] M. N. Schmidt and H. Laurberg. Non-negative matrix factorization with gaussian process priors. *Computational Intelligence and Neuroscience*, 2008.

[SLK+08] R. Schachtner, D. Lutter, P. Knollmüller, A. M. Tomé, F. J. Theis, G. Schmitz, M. Stetter, P. Gómez-Vilda, and E. W. Lang. Knowledge-based gene expression classification via matrix factorization. *Bioinformatics*, 24:1688–1697, 2008.

[SM07] T. T. Shannon and J. McNames. Ica based disturbance specific control charts. In *IEEE International Conference on Information Reuse and Integration*, pages 251 – 256, 2007.

[SMR+02] K. R. Skinner, D. C. Montgomery, G. C. Runger, J. W. Fowler, D. R. McCarville, T. R. Rhoads, and J. D. Stanley. Multivariate statistical methods for modeling and analysis of wafer probe test data. *Semiconductor Manufacturing, IEEE Transactions on*, 15:523 – 530, Nov 2002.

[SPL09] R. Schachtner, G. Pöppel, and E. W. Lang. Binary nonnegative matrix factorization applied to semi-conductor wafer test sets. In *ICA '09: Proc. of the 8th International Conference on Independent Component Analysis and Signal Separation*, pages 710–717. Springer, 2009.

[SPL10] R. Schachtner, G. Pöppel, and E. W. Lang. Nonnegative matrix factorization for binary data to extract elementary failure maps from wafer test images. In *Advances in Data Analysis, Data Handling and Business Intelligence*. Springer, Heidelberg-Berlin, 2010.

[SPST07] M. Stritt, G. Pöppel, and L. Schmidt-Thieme. Combining multi-distributed mixture models and bayesian networks for semi-supervised learning. In *IEEE International Conference on Machine Learning and Applications (ICMLA'07)*, 2007.

[SPTL09] R. Schachtner, G. Pöppel, A. M. Tomé, and E. W. Lang. Minimum determinant constraint for non-negative matrix factorization. In *ICA '09: Proc. of the 8th International Conference on Independent Component Analysis and Signal Separation*, pages 106–113. Springer, 2009.

[SRS08] M. Shashanka, B. Raj, and P. Smaragdis. Probabilistic latent variable models as non-negative factorizations. *Computational Intelligence and Neuroscience*, 2008, 2008.

[SS02] B. Schölkopf and A. J. Smola. *Learning with Kernels*. The MIT Press, 2002.

[SS05] F. Sha and L. K. Saul. Real-time pitch determination of one or more voices by non-negative matrix factorization. In *Proc. NIPS2004*, pages 1233–1240. MIT Press, 2005.

[SSU03] A. I. Schein, L. K. Saul, and L. H. Ungar. A generalized linear model for principal component analysis of binary data. In Frey (eds) Bishop, editor, *Proceedings of the 9th international workshop on artificial intelligence and statistics*, 2003.

[ST09] U. Schlink and A. Thiem. Non-negative matrix factorization for the identification of patterns of atmospheric pressure and geopotential for the northern hemisphere. *International Journal of Climatology*, 2009.

[STP+05] K. Stadlthanner, F.J. Theis, C.G. Puntonet, J.-M. Górriz, A.M. Tomé, and E.W. Lang. Hybridizing sparse component analysis with genetic algorithms for blind source separation. In *ISBMDA 2005. LNCS (LNBI)*, volume 3745, pages 137 –148. Springer, Heidelberg, 2005.

[SWH09] M. N. Schmidt, O. Winther, and L. K. Hansen. Bayesian non-negative matrix factorization. In *ICA '09: Proc. of the 8th International Conference on Independent Component Analysis and Signal Separation*, pages 540–547. Springer, 2009.

[TF09] V. Y. F. Tan and C. Févotte. Automatic relevance determination in nonnegative matrix factorization. In *Proc Workshop on Signal Processing with Adaptative Sparse Structured Representations (SPARS'09)*, 2009.

[Tip99] M. E. Tipping. Probabilistic visualisation of high-dimensional binary data. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 592–598, July 1999.

[TKGB99] K. W. Tobin, T. P. Karnowski, S. S. Gleason, and P. O. Box. Using historical wafermap data for automated yield analysis. In *11 th Annual Symposium on Electronic Imaging Science & Technology*, pages 23–29, 1999.

[Tro03] J. A. Tropp. Literature survey: Non-negative matrix factorization. available online at *http://www.acm.caltech.edu/ jtropp/notes/Tro03-Literature-Survey.pdf*, 2003.

[TST05] F. J. Theis, K. Stadlthanner, and T. Tanaka. First results on uniqueness of sparse non-negative matrix factorization. In *EUSIPCO*, 2005.

[Tur95] P. Turney. Data engineering for the analysis of semiconductor manufacturing data. In *In Proc. of the IJCAI-95 Workshop on Data Engineering for Inductive Learning*, pages 50–59, 1995.

[VCG08] T. O. Virtanen, A. T. Cemgil, and S. J. Godsill. Bayesian extensions to nonnegative matrix factorisation for audio signal modelling. In *Proc. of IEEE ICASSP 08*, 2008.

[vZ90] P. van Zant. *Microchip fabrication : a practical guide to semiconductor processing.* McGraw-Hill, 1990.

[Wan09] C.-H. Wang. Separation of composite defect patterns on wafer bin map using support vector clustering. *Expert Systems with Applications*, 36(2, Part 1):2554 – 2561, 2009.

[WGB+99] B. M. Wise, N. B. Gallagher, S. W. Butler, D. D. White, and G.G. Barna. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. *Chemometrics*, 13:379–396, 1999.

[WJ04] Y. Wang and Y. Jia. Fisher non-negative matrix factorization for learning local features. In *In Proc. Asian Conf. on Comp. Vision*, pages 27–30, 2004.

[WM96] S. Waterhouse and D. J. C. MacKay. Bayesian methods for mixtures of experts. *Advances in Neural Information Processing systems*, 8, 1996.

[WP07] O. Winther and K. B. Petersen. Bayesian independent component analysis: Variational methods and non-negative decompositions. *Digit. Signal Process.*, 17(5):858–872, 2007.

[XLG03] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273, New York, NY, USA, 2003. ACM.

[XYF08] S. Xie, Z. Yang, and Y. Fu. Nonnegative matrix factorization applied to nonlinear speech and image cryptosystems. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 55:2356–2367, Sept. 2008.

[YFH09] S. S. Young, P. Fogel, and D. M. Hawkins. Clustering scotch whiskies using non-negative matrix factorization. online at http://niss.org/sites/default/files/ScotchWhisky.pdf, 2009.

[ZC06] R. Zdunek and A. Cichocki. Non-negative matrix factorization with quasi-newton optimization. In *ICAISC*, pages 870–879, 2006.

[ZC07] R. Zdunek and A. Cichocki. Nonnegative matrix factorization with constrained second-order optimization. *Signal Process.*, 87(8):1904–1916, 2007.

[ZC08] R. Zdunek and A. Cichocki. Fast nonnegative matrix factorization algorithms using projected gradient approaches for large-scale problems. *Intell. Neuroscience*, 2008:1–13, 2008.

[ZDLZ07] Z. Zhang, C. Ding, T. Li, and X. Zhang. Binary matrix factorization with applications. In *Seventh IEEE International Conference on In Data Mining*, pages 391–400, 2007.

[ZYZ07] Z. Zheng, J. Yang, and Y. Zhu. Initialization enhancer for non-negative matrix factorization. *Eng. Appl. Artif. Intell.*, 20(1):101–110, 2007.

[ZZC06] D. Zhang, Z.-H. Zhou, and S. Chen. Non-negative matrix factorization on kernels. In *LNCS*, volume 4099, pages 404–412. Springer, 2006.